

USAISEC

US Army Information Systems Engineering Command
Fort Huachuca, AZ 85613-5300

4

U.S. ARMY INSTITUTE FOR RESEARCH
IN MANAGEMENT INFORMATION,
COMMUNICATIONS, AND COMPUTER SCIENCES
(AIRMICS)

FILE COPY

AD-A216 807

INTEGRATED OFFICE INFORMATION SYSTEM (IOIS)

(ASQBG-A-89-030)

June, 1989

DTIC
ELECTED
JAN 17 1990
S B D

AIRMICS
115 O'Keefe Building
Georgia Institute of Technology
Atlanta, GA 30332-0800



DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

90 01 17 005

This research was sponsored by the Army Institute for Research in Management Information, Communications, and Computer Sciences (AIRMICS), the RDTE organization of the U.S. Army Information Systems Engineering Command (USAISEC). The purpose of this research is to study and build a prototype of an Integrated Office Information System (IOIS). This research report is not to be construed as an official Army position, unless so designated by other authorized documents. Material included herein is approved for public release, distribution unlimited. Not protected by copyright laws.

THIS REPORT HAS BEEN REVIEWED AND IS APPROVED *for*

Approved	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
A-1	

s/ James Gantt
James Gantt
Chief, Management Information
Systems Division

John R. Mitchell
John R. Mitchell
Director
AIRMICS



Integrated Office Information System (IOIS)†

Dr Jay F. Nunamaker

Dr Olivia R. Liu Sheng

Milam W. Aiken

Chandra S. Amaravadi

Kunihiko Higa

Charles M. Morrison

Luvai Motiwalla

Department of Management Information Systems

College of Business and Public Administration

University of Arizona

Tucson, Arizona 85721

602-621-2748

June 15, 1989

†Submitted to the Army Institute of Research in Management Information, Communications and Computer Science (AIRMICS)

Abstract

The objective of this research is to study and build a prototype of an Integrated Office Information System (IOIS). The need for this system is derived from problems in existing automated office systems including: the lack of standard software tool user interfaces, excessive use of paper to transmit information, redundant and inconsistent information in common databases, and haphazard coordination and control of information exchange. The proposed IOIS will address these problems by providing the means to accomplish a variety of office tasks while maintaining data integrity, reducing the flow of paper, improving communication, and standardizing user interfaces and software protocols. *INTEGRATED SYSTEMS, INFORMATION RETRIEVAL*

Efforts were centered on analyzing processes in AIRMICS and finishing the detailed design of the proposed architecture. In addition, the following processes were implemented during the current contract period:

1. A key word help facility for describing projects and for storing and retrieving project histories.
2. A personnel allocation module that uses personnel profiles for optimal assignments.
3. An Expert Session Planner (ESP) for planning GDSS sessions.
4. A calendar/scheduler to monitor project activities, schedule GDSS sessions, and make appointments.

The primary focus for July 89 - December 89 will be to move portions of the IOIS into the UNIX environment. The interface and parts of RME will be ported to UNIX while the GDSS tools will be left running under DOS. These DOS applications can be accessed through UNIX using AT&T's Simul-Task 386 software. The January 90 - December 90 period will concentrate on designing and building a distributed KB/DB for the IOIS.

The proposed project will be completed over 18 months divided into two periods. The first six month period will require a budget of \$75,102. The following twelve month period will require a budget of \$150,866.

Table of Contents

1	Introduction	5
2	Objectives Met	6
2.1	Goals	6
2.2	Initial Architecture	6
2.3	AIRMICS Analysis	6
2.4	Current Design	7
2.5	Hardware and Software	7
2.6	Contributions	7
3	Analysis	8
3.1	Case Study	9
3.1.1	Research Method	9
3.1.2	Organizational Structure	9
3.1.3	Background Information on AIRMICS	12
3.1.4	General Descriptions of Processes in AIRMICS	12
3.1.5	Detailed Descriptions of AIRMICS Processes	19
3.1.6	Conclusion	26
3.2	Hardware/Software Selection	30
3.3	Nexpert Review	31
3.3.1	Why Nexpert?	31
3.3.2	Product Support	31
3.3.3	Learning Nexpert	31
3.3.4	Nexpert's Features	32
3.3.5	Nexpert Pros and Cons	35
4	Design	36
4.1	AGDSS	36
4.1.1	Knowledge Acquisition	36
4.1.2	Knowledge Representation	37

4.1.3	Inferencing Strategy	38
4.2	RME	38
4.3	IOIS Database and Knowledge Base Design	51
4.3.1	Introduction	51
4.3.2	Structured Entity Model (SEM)	52
4.3.3	Description of Database Schema	52
4.4	Knowledge Base Design Using SEM	55
4.4.1	Interface	67
4.4.2	Calendar / Scheduler	68
5	Implementation	68
5.1	AGDSS	69
5.2	ESP Implementation Status	72
5.3	Implementation of RME	73
5.3.1	Keyword Help	74
5.3.2	Retrieve Project History	75
5.3.3	Retrieve Project History	75
5.3.4	Personnel Allocation	75
5.3.5	Prepare the Acquisition Package	76
5.3.6	Resource Status/Update	76
5.4	Calendar/Scheduler	78
6	An AIRMICS Scenario Illustrating the Use of IOIS	78

List of Figures

1	The organization of the Army	10
2	The Organization of AIRMICS	13
3	Overview of AIRMICS processes	20
4	Allocating the Resources	22
5	Administering Contracts	23
6	Preparing the Acquisition Package	25
7	Monitoring the contract	27
8	Contract Completion	28
9	Architecture of IOIS	39
10	Architecture of RME	41
11	The Project Planning Component of RME	44
12	The Resource Management Component of RME	47
13	The Contract Administration component of RME	49
14	A Prototype IOIS Data and Knowledge Structure	53
15	The SEM Diagram for Keyword Search Function.	56
16	Inference Patterns for Keyword Search.	57
17	Transformation of SEM Entities to Nexpert Classes.	59
18	A Single Class Rule-net.	60
19	A Subclass Rule-net.	61
20	A One-to-many Rule-net.	62
21	A Many-to-many Rule-net with One Inference Pattern.	63
22	A Many-to-many Rule-net with Two Inference Patterns.	64
23	A Combination Rule-net: One-to-many with Subclass.	65
24	A Combination Rule-net: Many-to-many with Subclass.	66
25	Architecture of AGDSS	69
26	ESP High-level Architecture	70
27	The Keyword Structure	74
28	Resource Allocation	76
29	Personnel Allocation Rules	77

30	An AIRMICS Scenario Using IOIS	79
31	AIRMICS Scenario Contd..	81
32	AIRMICS Scenario Contd..	82

1 Introduction

The Integrated Office Information System (IOIS) architecture is an advanced office architecture being developed for the U. S. Army. The project is being funded by the Army Institute of Research in Management Information Communications and Computer Science (AIRMICS).

The development has been motivated by the absence of integrated office frameworks for the office. Existing frameworks consist of procedural and form automation systems for non-managerial office workers, and generic tools (such as spreadsheets) for managerial and professional personnel. Concepts to support different levels of workers, such as tailoring generic tools for office environments, using common interfaces, and providing AI techniques have not yet been addressed within a single framework. Reasons for developing an integrated architecture rather than using isolated systems for clerical and managerial workers include: an integrated architecture will reduce maintenance and allow better control of office and system operations, sharing of common data and knowledge bases will contribute to reduced redundancy and improve application development time, and common interfaces will reduce training costs.

The components for the IOIS architecture were selected based on literature stating the requirements of managerial, professional and clerical personnel [2, 3]. For managers, providing support for decision making, meetings, and resource management was considered important, while professional and clerical workers primarily required access to databases, form management systems, and conventional office tools. The initial IOIS framework includes a resource management expert, an asynchronous GDSS facility, some decision support systems, a calendar/scheduler, and some conventional office tools with a common knowledgebase/database and interface providing the integration.

A study of the AIRMICS office was conducted to provide a target implementation environment. Currently, part of the architecture is being implemented using Neuron Data's "Nexpert Object", an expert system shell. The implementation, completed in late June 1989, includes the KB/DB, the Resource Management Expert, an Expert Session Planner and Session Facilitator for carrying out GDSS sessions, and an intelligent calendar/scheduler. Research is also being conducted to refine the architecture.

The project was supervised by Dr. Olivia Sheng and carried out by four doctoral students. Several masters student's projects related to IOIS have also been completed.

2 Objectives Met

2.1 Goals

The major goals of the project were: to develop an architecture allowing integration of customized office applications and commercial applications, to provide support to all levels and types of office workers, to present common interfaces, to adjust interfaces to user preferences and access levels, and to enable users to share common office knowledge and data. These goals were then to be translated into a detailed design and implemented using state-of-the-art tools and techniques.

2.2 Initial Architecture

An initial architecture, based on a case study at the University of Arizona's MIS department office, was proposed in March 1988. It consisted of an interface module, a diagnostic module, a KB/DB component, and support tools. The interface module made use of user profiles to screen users, and the diagnostic module used tool and problem profiles to diagnose the user's problem and present the required tool for solving the problem. The KB/DB was required for the purpose of maintaining office data and knowledge such as procedures, policies, regulations, personnel, etc.

There were two problems with the early prototype: the needs of the MIS office were not sufficient to demonstrate the full capability of IOIS, and feedback received from AIRMICS personnel indicated that the tools should be tailored specifically to the Army environment. An attempt to study the SID office at Ft. Huachuca did not succeed due to security regulations that prevented the researchers from obtaining the detailed data required for developing the IOIS prototype.

2.3 AIRMICS Analysis

An attempt was then made to obtain an analysis of the AIRMICS office. In late summer 1988, the ISP study of AIRMICS information flows conducted by students at Georgia Tech became available. This study provided necessary background information for an AIRMICS study subsequently conducted with the help of Major Ted Hengst. This study identified the data flows, organizational structure, and functional allocations in AIRMICS.

2.4 Current Design

Using this analysis, the researchers developed detailed designs for the Interface, Resource Management Expert (RME), Expert Session Planner (ESP), Expert Session Facilitator (ESF), and the Knowledge Base / Data Base (KB/DB). The interface was improved after an extensive literature review, and now features groupings of menu selections based on tasks, security screening, and nested menus that return users to the point of selection. Because of the a close inter-relationship between the requirements and the tools, the diagnostic module has not been included in the current design. An application monitor has been proposed for saving current application information: for example, if a user was interrupted during an editing session, the application monitor would remind him of the unfinished task when he returned. The knowledge base/data base has been greatly revised and improved using the SEM methodology.

2.5 Hardware and Software

Appropriate hardware and software were selected and acquired and a subset of the revised design has been implemented. The prototype was developed on an AT&T 6386 personal computer with a 135 MB hard disk, and 4 MB RAM. An expert system shell, "Nexpert Object," by Neuron Data, provided the environment for the knowledge base and data base, and "Vermont Views" and "Windows for C" were used to build the interface. The implementation includes the Resource Management Expert, Expert Session Planner (for planning GDSS sessions), a multi-criteria group decision making tool, an intelligent calendar/scheduler, and access to conventional office tools. The prototype was targeted at supporting activities considered most important for AIRMICS.

A summary of the project history appears in *Table 1*.

2.6 Contributions

The IOIS design identified the features of an integrated office system, and laid groundwork for knowledge-based approaches. For example, the resource management function is important to AIRMICS and to office systems support in general, therefore, an expert system-based resource allocation tool has been implemented.

IOIS also incorporates expert systems technology while accessing commercial tools. The prototype made use of Nexpert features such as accessing to databases, repeatedly firing sets

Mar 1988	Pilot system for IOIS.
Aug 1988	IOIS Architecture completed.
Sep 1988	First version of case study, hardware requirements.
Oct 1988	Received Nexpert 1.0.
Jan 1989	Developed SEM methodology for integration of DB/KB.
Jan 1989	Received version 1.1.
Feb 1989	ESP tool selection.
Feb 1989	ESP group selection.
Mar 1989	Mike attended Bechtel seminar.
Apr 1989	Mike attended user group meeting.
Apr 1989	Completed keyword help.
Apr 1989	Personnel allocation.
Apr 1989	Text based interface designed.
Apr 1989	Distributed EBS and IA.
Jun 1989	Final prototype completed.

Table 1: Project Schedule in Retrospect

of rules, dynamically (inference controlled) navigating object networks, certainty factors, and callable interface.

3 Analysis

The purpose of the analysis phase was to understand the requirements of AIRMICS, to understand the design and capabilities of tools within the IOIS architecture, and to identify a suitable hardware/software environment for the implementation. The design and capabilities of tools within the IOIS architecture have been summarized and maintained in a previous report. The case study and the identification of a suitable implementation environment are discussed below.

3.1 Case Study

A case study of the MIS office was conducted in the early spring of 1988 and a prototype system was subsequently developed. The feedback obtained from AIRMICS personnel after the IOIS project review in March 1988 revealed that substantial differences exist between the MIS office and AIRMICS, the intended target for the IOIS prototype. An attempt was made to study the SID office at Ft Huachuca, but it did not succeed due to security regulations preventing the researchers from obtaining the detailed data required. Meanwhile, the study of AIRMICS information flows conducted by students at Georgia Tech became available to the researchers. This provided background information required for the study of AIRMICS that is described below.

3.1.1 Research Method

A Business Systems Planning (BSP) study was conducted by master's students at the Georgia Institute of Technology in the Spring of 1988 [4, 5]. This study, referred to as Information Systems Plan (ISP), identified AIRMICS' major processes, information flows, and their relationships. Major functions related to the mission of AIRMICS were identified and formed the focus for the study described here. With a preliminary understanding of the major functions, structured interviews were conducted with Major Ted Hengst of AIRMICS. The information about the organizational structure of the Army was obtained from interviews with Mr. Joseph Tuggle, at Systems Integration Directorate (SID), and Captain Lee Price, Systems Engineering Directorate (SED), at Ft. Huachuca.

3.1.2 Organizational Structure

This section describes the relationship of AIRMICS with respect to the different branches of the Army and is illustrated by the organizational chart in *Figure 1* and described below.

The Army is organized into five Major Army Commands (MACOMs): FORSCOM, ISC, TRADOC, AMC and USAREUR. These are under the command of the Department of Army Headquarters (HQDA). Staff agencies such as the DCS Plans (Deputy Chief of Staff - Plans) oversee all planning with respect to the MACOMs. Other staff agencies at the same organizational position as DCS Plans include DCS Ops (Operations), DCS Logs (Logistics), DCS (Personnel).

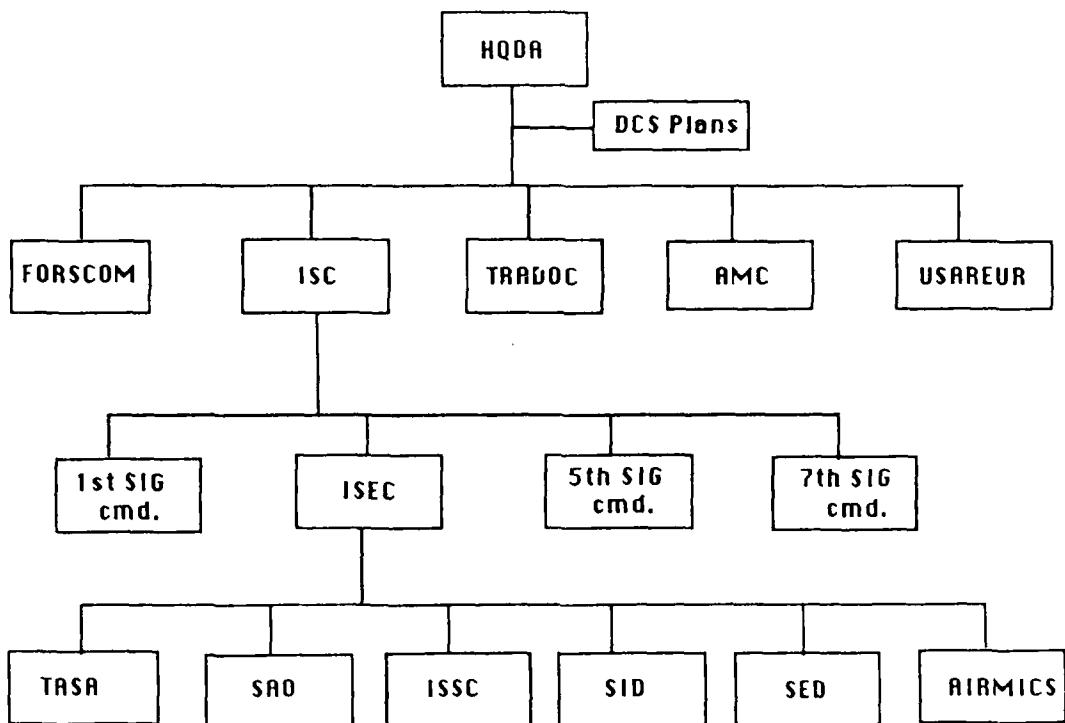


Figure 1: The organization of the Army

These agencies also exist at each organizational level (not shown) and function to develop plans, coordinate contracts, installations etc.

The functions of the MACOMs are described as follows:

FORSCOM - The Forces Command (FORSCOM) is the organization of personnel and troop commands that have combat and combat support roles located in CONUS (Continental United States). FORSCOM is organized into units known as Corps, which are divided into divisions. The divisions are divided into command brigades, which are again divided into battalions and further into companies. The communications requirements of these units and subunits are met by ISC (described below).

ISC - The Information Systems Command (ISC) is responsible for all communications systems in the Army. The emphasis is on systems for gathering strategic information from the corps down to the company level. The communications hardware is developed/supplied by the AMC while ISC develops and maintains the software. ISC is usually associated with the larger systems rather than with those at the company level, and is not concerned with

health and safety systems.

TRADOC - Training and Doctrine Command. The TRADOC has responsibility for all Army training. It sets the doctrine or policy to be followed by all divisions of the Army, develops training requirements, and makes future policy.

AMC - The Army Material Command (AMC) is responsible for developing the hardware and other materials required for all Army weapons and communication systems. It is organized into Program Executive Offices (PEOs), each carrying out the development in a specific area (e.g Automotive vehicles). The PEOs have Project Managers (PMs) for functions such as ensuring parts availability.

USAEUR - This is the European wing of the combat forces of the U.S. Army.

ISC is divided into four major sub units: The 1st, 5th, and 7th signal commands and ISEC. The signal commands are operational units that have geographical responsibility for maintaining communications and automation equipment:

1st Signal Command - responsible for the Pacific

5th Signal Command - responsible for Europe

7th Signal Command - responsible for the continental United States.

ISEC - The Information Systems Engineering Command (ISEC) is an agency responsible for supporting the development and installation of worldwide information systems. It provides support for communications, visual information, records management, office automation and printed publications. In addition, ISEC supports special purpose software. It is split up regionally into centers based on where the services were needed. Both contract and in-house development are used. ISEC is organized into the following divisions:

TASA - Training and Support Activity (TASA) for troops. This includes visual information and printed publications regarding training, regulations etc.

SAO - Systems analysis office (SAO) provides simulation and analysis for the other Army units in areas such as communications.

ISSC - Information systems software centers which fulfill the software development needs within ISC. There are five different software centers: Ft. Lee, Washington D.C., Atlanta¹, Ft. Ben Harrison, and Ft. Huachuca.

SID - The System's Integration Directorate (SID) conducts the remainder of the engineering associated with completing a system that has been approved for installation. SID is responsible for investigating the technology and for providing the technical input to the DCS Plans that exists at the directorate for ISC.

SED - The System's Engineering Directorate (SED) provides engineering for communication and automation for information systems.

AIRMICS - AIRMICS provides inputs to the technology acquisition process. The mission of AIRMICS is to sponsor and conduct applied research in the areas of telecommunications, automation, audiovisual, record management and publications systems in support of USAISC needs. It serves as a clearing house for technical information to ISC as well as to other units within the Department of Defense.

3.1.3 Background Information on AIRMICS

AIRMICS is located in the campus of the Georgia Institute of Technology in Atlanta, Georgia. It presently it has 20 employees including a director, three division chiefs, 14 researchers, a budget analyst and a secretary. AIRMICS is organized into three divisions: the Communications and Network System's Division (CNSD), Computer and Information Systems Division (CISD) and the Advanced Concepts and Technology Integration Division (ACTID). This is illustrated in *Figure 2*. The office of the director (Admin) controls and administers the AIRMICS organization. The different divisions are housed within the same building. It is interesting to note that AIRMICS as well as its divisions satisfy the definition of an office by having at least two hierarchical levels and four personnel.

3.1.4 General Descriptions of Processes in AIRMICS

To realize its mission of conducting and sponsoring research, AIRMICS carries out a number of research, financial, administrative, and management functions. This study focuses on those

¹This may not be a software center in the future

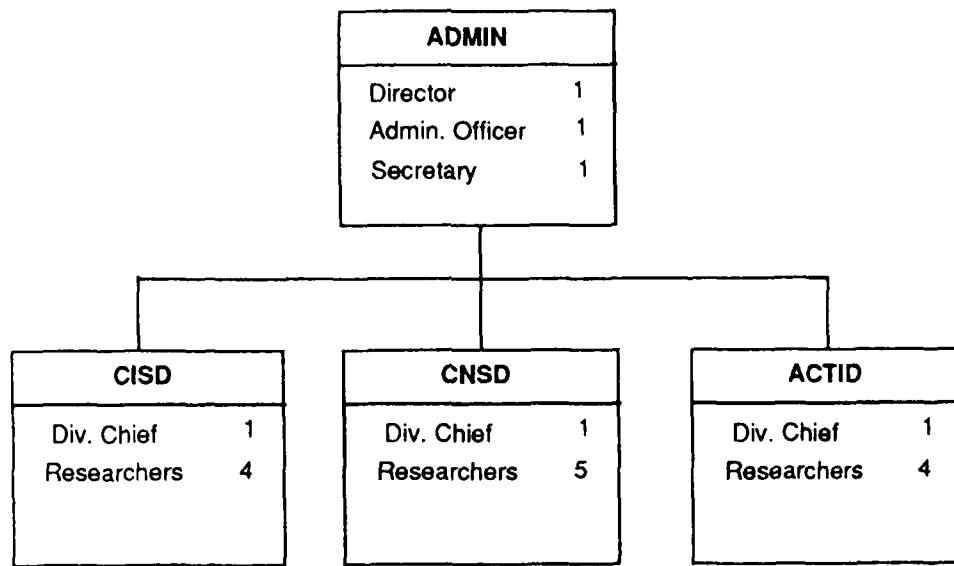


Figure 2: The Organization of AIRMICS

functions and procedures of AIRMICS related to initiating, monitoring and closing contracts or research projects. For convenience, these functions will be described in a sequence, although this is not necessarily what really happens:

1. The process starts with the identification of areas of interest.
2. This is followed by a decision on the projects to fund.
3. Next a "procurement package" package is prepared for conducting the research.
4. A contractor is selected and the contract is awarded.
5. The project is monitored until ...
6. Completion.

There are variations on this sequence, and some of the steps may be omitted. Now these functions will be explained in detail:

Since AIRMICS conducts and sponsors research, it develops short- and long-term research plans. The function of identifying research areas and conducting research is referred to as RD&TE

Table 2: Interest Areas of AIRMICS

No.	Area	Division
1	Supercomputers	CISD
2	Communications and network technology	CNSD
3	Distributed computer systems	CNSD
4	Very large databases	CISD
5	Software engineering	CISD
6	Decision Support Systems	ACTID
7	Information security	ACTID
8	MIS	ACTID
9	Artificial Intelligence	all three
10	Technology transfer	all three

(Research, Development and Technical Evaluation). For the short term, the plan spans the current year, while for the long term the planning horizon is 5 years. The short-term plan includes the budget for specific research areas and is developed before the fall of the year preceding the year to be budgeted; the budget for 1989-1990, for instance, is developed before December of 1988. AIRMICS has a yearly budget of \$2 million divided between funded research and operating expenses. The organization may receive additional funding from other organizations within the DOD such as AWIS (Army Wide Information Systems) and the Technical Directorate at Ft. Huachuca PEO/PMs for research in a specific area. Additional funding may be as high as \$2 million every year.

Plans are developed based on the available funding and the areas of interest. RDTE is divided into ten areas that are of interest to AIRMICS. These areas reflect both historical tradition as well as the requirements of the Army as a whole. They are updated based on the army requirements, compilation of the research results, and on the available technology. The responsibility for research in these areas is distributed among the three divisions. The RDTE areas and the divisions responsible for them are shown in *Table 2*.

RDTE plans are based on continuing work in areas of interest, in beginning work in new areas that are revealed as a result of prior research, or on specific research requirements of another branch of the DOD. Thus all research that is conducted in-house or contracted with another organization has to conform to the areas of interest interest in the RDTE plan. AIRMICS solicits sponsorship from other agencies within the DOD for research on areas of interest or in related areas, and these additional funds are worked into the overall plan. For example, if "security issues in large databases" have been identified as an area of interest and \$50,000 was allocated for it, and the Navy also requires research in the same area and provides an additional \$50,000 for research in this area, AIRMICS has \$100,000 for this research However, it also has the option of spending only \$50,000 and utilizing the additional funds for other research.

The RDTE five year plan is reviewed by an administrative board consisting of the Technical Director of ISC, Commander of ISEC, Technical Director of ISEC, Director of DCS Plans, and the Director of AIRMICS. From this process, a list of tasks to be included in the current research plan is developed. The decision regarding conducting the research in-house versus contracting it out to an external contractor depends on the availability of expertise and personnel in-house. A contracting officer (COR) is appointed by the AIRMICS director and the division chiefs to monitor the project. The selection is based on the expertise required for the project. For large projects, more than one COR may be designated. The function of the COR is to manage the project and to act as a liaison between the contractor and AIRMICS and between AIRMICS and the contract/finance office in Ft. McPherson. He mainly reports to his division chief. For an internal project, research personnel are also allocated to the project at this stage. Researchers typically hold advanced degrees in Computer Science, Electrical Engineering, Operations Research, or MIS, and execute research tasks such as identifying and developing concepts, carrying out the analysis and design, and implementing the concepts. Projects are always in process at AIRMICS; during a typical year, 10-15 projects may be underway. When projects terminate, the results are made available to the rest of the DOD organization. In some cases AIRMICS may directly provide copies of the reports to interested organizations, and thus serves as a clearing house for technology.

As previously stated, research in areas of interest is either conducted in-house or contracted to another organization (referred to as the contractor). There are four methods of issuing contracts:

Competitive bids - AIRMICS completes a work order and submits it to the contracting offices of the DOD. The DOD solicits bids from contractors, and these are evaluated through a competitive process. The bidding and evaluation process is slow and may take more than a year.

Broad agency announcement (BAA) - An advertisement is placed in the Commerce and Business Daily for the research to be conducted. Contractors then submit proposals which are evaluated. Sometimes contractors may opt to submit white papers or "concept papers" to elicit preliminary feedback. The BAA process can take between 6 weeks to 9 months depending on the size of the contract.

8A task order - The government requires a certain percentage of the expenses, incurred by any DOD organization on external contracts, to be contracted to minority owned/operated organizations. AIRMICS maintains contact with a few of these to contract out tasks so as to meet the DOD requirements. As an example, the Information System's Network organization in Bethesda was recently employed for upgrading certain video equipment. (The "8A" refers to the form used to obtain contracts of this type).

Georgia Tech task order contract - AIRMICS has an ongoing agreement with the Georgia Institute of Technology for carrying out specific research tasks.

The BAA is the vehicle most frequently used by AIRMICS to acquire the services of a contractor because of its faster turnaround time.

After identification of the research projects and the allocation of the personnel, the next step for external projects is to develop an acquisition package. This consists of a technical evaluation of the project, a statement about the uniqueness of the project, a Purchase Requisition and Commitment (PR& C) form required for the BAA process, a Work Unit Summary Form, and the project proposal submitted by the contractor.

To assess the uniqueness of the project, the COR performs an online search of prior research conducted in the DOD. A database search form is filled and sent to the DTIC (Defense Technology Information Center) information center in Cameron, New Jersey. AIRMICS has a Unix access package (OLE) to access the information center; the center has an online repository of all research conducted by the different branches of the DOD. Filling the search form consists of identifying

and keying in the keywords describing the project from a reference manual. The manual specifies the keywords that must be used for each topic that describes the project. The DTIC process ensures that the research was not already been conducted or sponsored by any DOD unit or government agency, and establishes the uniqueness of and need for the project.

An acquisition package is prepared that contains the project proposal, a project evaluation form and a completed search form certifying that the research was not already conducted. The project evaluation form is a statement of the research value and technical viability of the project; a PR & C form is also included which contains the names, addresses and phone numbers of the principal parties to the contract, namely the COR, his division chief, the director, the budget analyst and the principal instigator of the project. The package also contains a summary of the contract — a description of the project and expected results.

The package is then forwarded to the contracting officer at Ft. McPherson. The contracting officer conducts a financial analysis of the contract and either approves it or suggests changes; changes are made by the Contractor and co-ordinated by the COR. The project is then formally awarded by obtaining signatures from all parties to the contract (on the PR & C form): the finance officer, the COR, the contractor, and the director of AIRMICS. Payments are issued to the contractor according to type of the contract. For example, some contracts specify payments are to be made contingent on the progress accomplished by the contractor, and the COR communicates approval of progress to the contract administration office. It must be noted that the COR does not take actions independently but is directed by the division chief or the director of AIRMICS. Upon obtaining approval to issue payment, the contract officer disburses the payment to the contractor.

After the project is selected and awarded to the Contractor, the project is "in process." The next phase of RDTE is managing the contract. The COR of the project co-ordinates periodic reviews with the contractor to ensure that the progress is on schedule. This may involve examining software development, providing information, or scheduling future reviews. These reviews are called IPRs (In Process Reviews).

A procedure involving a TDY (Temporary Duty) form is executed by the COR to travel to the site of the contractor. The travel form is signed by the COR's division chief and sent to the contract/finance office in Fort McPherson for estimating/verifying the expenses claimed. If

a travel advance is requested, the office issues a voucher for the payment. Up to 40% of the total expenses may be issued in advance. TDY forms may also be filled by researchers at AIRMICS for attending seminars or for obtaining training. If the training is within the DOD, then a 1598 form (an intra-Army accounting tool) is filled and sent to Fort McPherson and the cost of the training is deducted from the AIRMICS account with Fort McPherson.

IPRs are conducted periodically. Research on projects may often necessitate changes; if the changes are suggested by the Contractor, they have to be approved by the COR. The Contractor is responsible for ensuring that the project is completed within budget and on schedule. In cases where the project cannot be completed within the deadline, a no cost extension of the due date may be granted by the COR with the approval of the contract administrator.

Equipment requirements are estimated as part of the contract by the Contractor for external projects or by the COR for internal projects. In the event that special equipment is required (internal or external) for a project, a purchase request is sent to Fort McPherson, where the request is checked for "sole sourcing." This process ensures that equipment is purchased only on the basis of requirements and that no single vendor is favored. A GSA (Government Supply Agency) form is filled and the purchase contract is either approved or cancelled. After a contract is completed, the equipment is either left with the Contractor or abandoned in place if it is at a different site. For internal research, each researcher has a workstation on his desk. If special equipment is needed, it is accessible in the equipment laboratory that has several workstations in a networking environment. Additional equipment facilities are available through a service contract with the Georgia Institute of Technology.

When the project terminates, a final IPR is conducted. The director, the division chief, the COR, and any interested party within AIRMICS may participate in the review. The final IPR may involve a software demonstration and/or a formal presentation of research results. The Contractor submits the completed software/hardware, technical reports and documents. In addition AIRMICS requires the contractors to submit all reports and documents on regular 5 1/4" floppy diskettes to minimize its photocopying costs. After the final IPR, the COR updates the DTIC database and performs some additional internal procedures to terminate the contract. If AIRMICS is interested in continuing the project, it may renew the contract for another period and the same process is repeated.

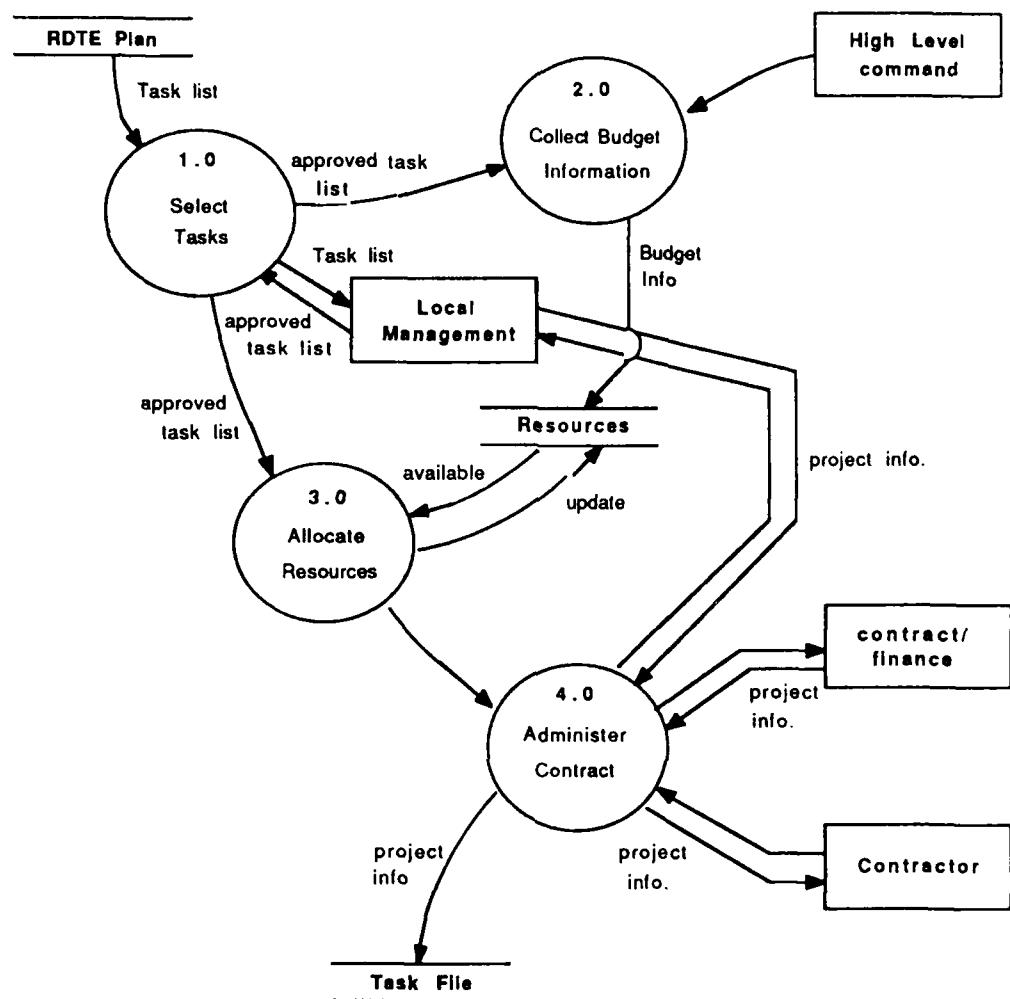
3.1.5 Detailed Descriptions of AIRMICS Processes

In this section, the AIRMICS processes are described with respect to the data flow diagrams in *Figures 3-8*.

Level 0 - The overall process: the overall process that AIRMICS goes through to plan, execute and evaluate research in the areas of interest is shown in *Figure 3*. The process consists of selecting tasks (1.0), collecting the budget information (2.0), allocating resources (3.0), and administering contracts (4.0). “Select tasks” receives a list of tasks from the data store, RDTE plan. This data store contains details of past projects, the interest areas of AIRMICS, the requirements of the Army, and perhaps, proposals from contractors. The director and the division chiefs review the task list and select the tasks/projects to be funded. This is the flow, “approved task list” from the entity, local management. The process “Collect budget information” (2.0) receives the approved task list and requests funding from the DOD (High level command). This is the budget information that goes to the data store, “Resources.” The resources include funds, manpower, and equipment. The process “allocate resources” (3.0), uses and updates this data store. It also uses the information, “approved task list.” At this point, decisions are made on the assignment of COR, equipment and research personnel. The process “administer contract” (4.0), involves the data store, “task file”, and two external entities, the contract/finance office at Fort McPherson and the Contractor. “Administer Contract” receives the progress reports, changes to the original contract/project, and the final project report from the Contractor and updates it to the task file. The progress reports, final project report and the reports of changes is the aggregate flow, “project information” in the diagram. “Administer Contract” communicates any changes to the contract/finance office and receives approvals for the changes if they involve change in deliverables or extensions of deadlines.

The reader should note that the processes are almost identical for internal and external projects with only slight differences. For instance, the COR of an internal project may not have to travel to the site of the Contractor since the Contractor is AIRMICS itself. But, the COR and the project team members may travel to conferences. The lower level processes are described below.

Level 1 - Selecting tasks: This process is carried out through a group meeting involving the director of AIRMICS, the division chiefs, the commander of ISC, the commander of ISEC and



An Office Study

Figure 3: Overview of AIRMICS processes

director of DCS Plans. As stated, it makes use of the task list and the interest areas from the store, RDTE plan and outputs an approved task list. This is however not decomposed further.

Level 2 - Collecting the budget information: The funding is provided by the DOD high level command, based on the approved task list and information on this resource is routed to the data store "resources," as described. This process is also not decomposed further.

Level 3 - Allocating resources: Resources consist of the funds, manpower and equipment. As stated earlier, the COR is appointed by the director and the division chiefs depending upon the area of expertise of the potential CORs and the expertise required by the project. Researchers are also similarly assigned to the project. The process "Allocate Resources" (3.0) is shown in *Figure 4*. It consists of the processes "assign personnel" (3.2) which uses and updates the data store "manpower," "allocate funds" (3.1) that uses and updates the data store funds, and the process "match contract specs with AIRMICS requirement specifications" (3.3). As mentioned earlier, the process of allocating funds (3.0) is done by the administrative board. This process communicates with the external entity, the contract/finance office in Ft. McPherson about the funds availability. The input is the flow, "budget line," and the output is the "allocation breakdown" which updates the store, "funds," after the process is completed. The process of matching Contractor specifications for equipment with AIRMICS equipment requirement specifications (3.3) is done by the COR. Equipment availability is checked (3.4) by searching the data store, "equipment" for the requirements. If the equipment is not available, the COR prepares the equipment specifications (specs) and obtains approval from the contract/finance office. As a result, new equipment may be leased or borrowed from the vendors and is updated in the "equipment" data store.

Level 4 - Administering contracts: This is the process that explodes into several levels and can be considered as a major activity of AIRMICS (*please refer to Figure 5*). After projects are selected for research funding in the RDTE plan, there are several processes involved, which have all been grouped under the process, "administer contracts" (4.0). "Administer contracts" consists of preparing the acquisition package (4.1), awarding the contract (4.2), monitoring the contract (4.3) and completing the contract (4.4). These processes are described below and are illustrated in *Figures 5-8*.

Level 4.1 - Preparing the acquisition package: The process of preparing the acquisition

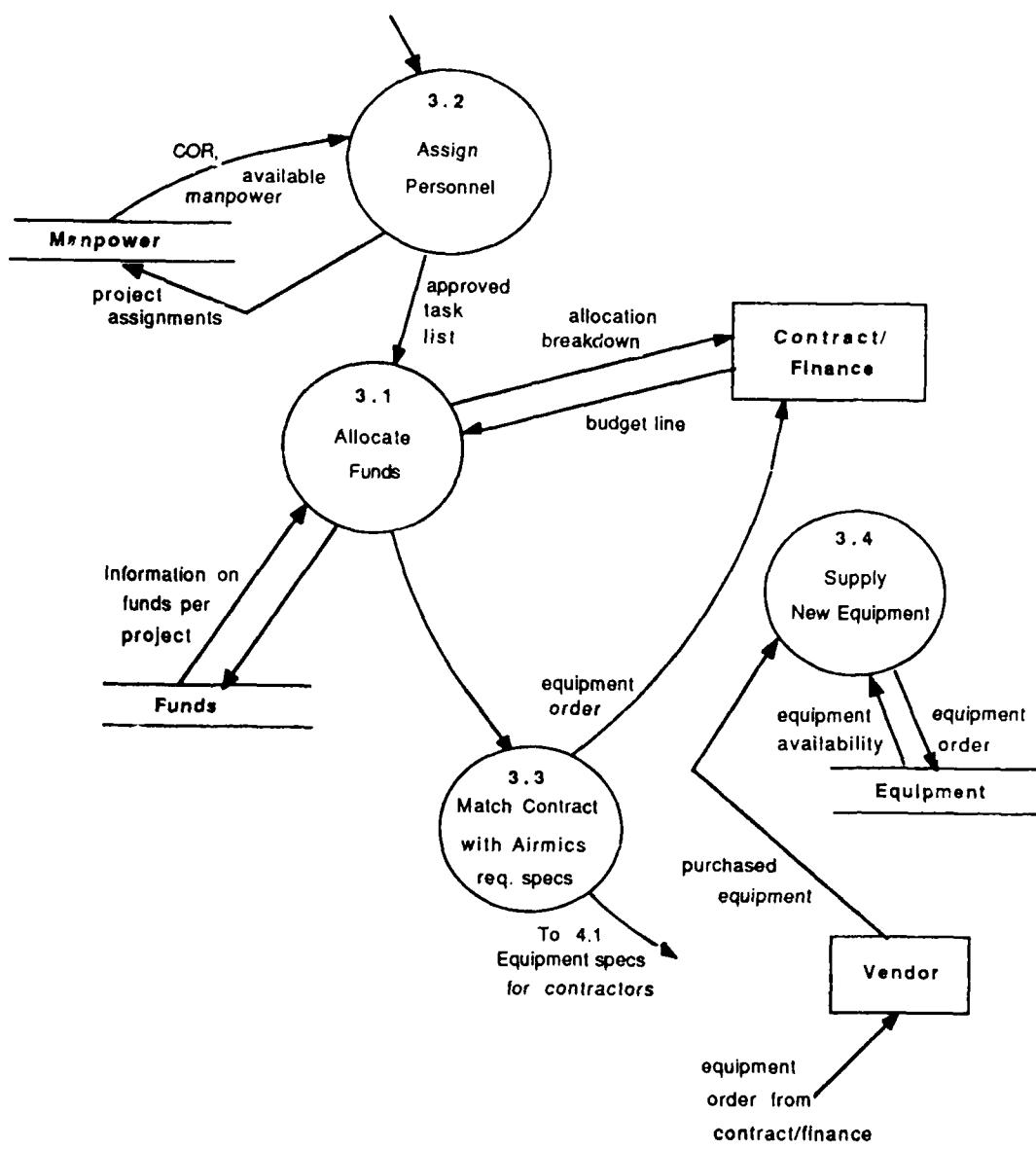


Figure 4: Allocating the Resources

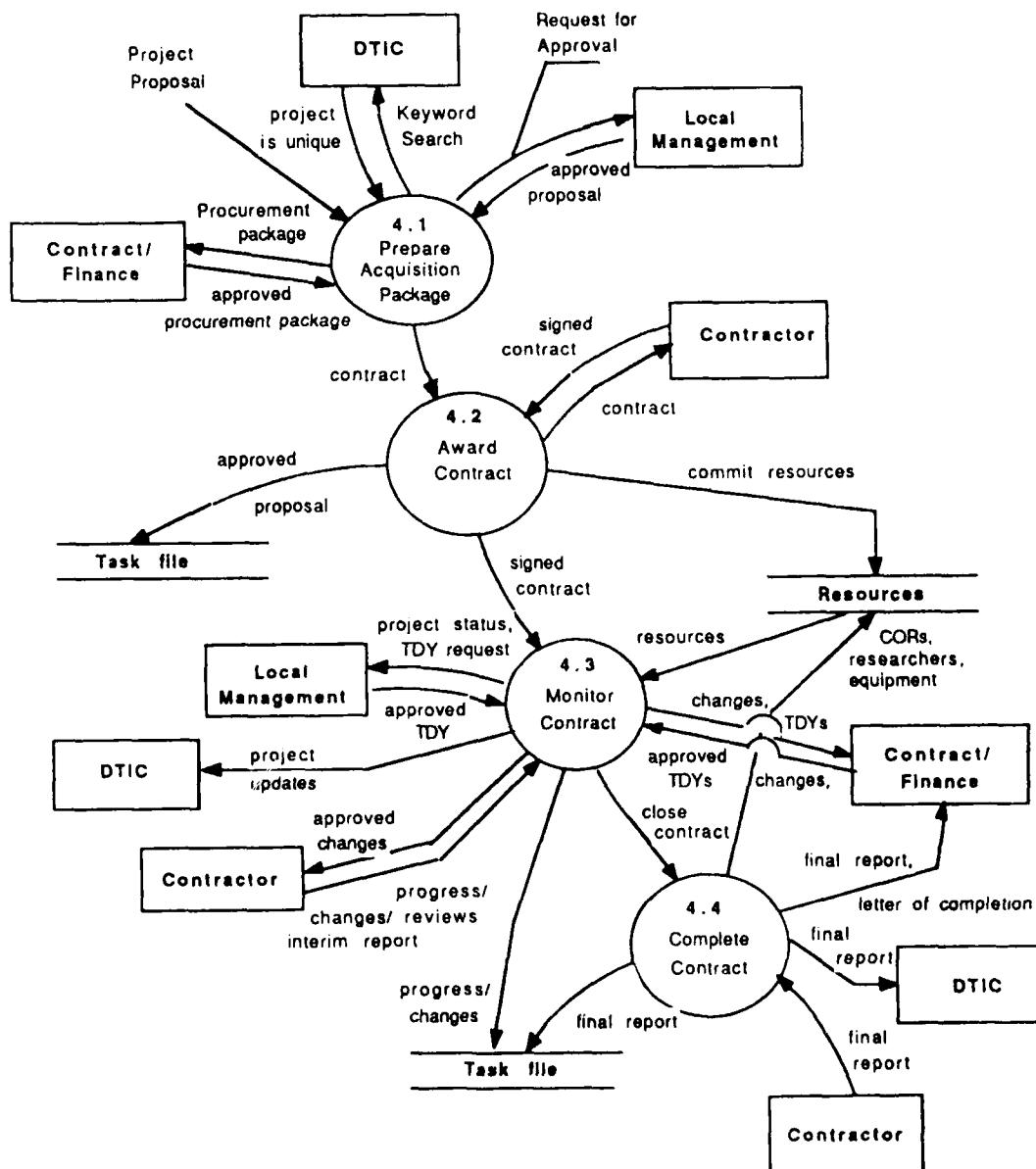


Figure 5: Administering Contracts

package (4.1) consists of conducting the DTIC search (4.1.1), evaluating the project (4.1.2), preparing the procurement package (4.1.3), and sending the procurement package (4.1.4) as illustrated in *Figure 6*. The DTIC search (4.1.1) consists of selecting the search conditions for the database from a reference manual and conducting the search with the OLE system. It uses the Contractor's proposal as input. The result of this process is a statement by the COR that the project is unique. If similar research was already accomplished and the results are available, the Contractor and the COR may together revise the proposal. The output of 4.1.1 are the project proposal and the uniqueness statement. After the DTIC search, the COR evaluates the technical content of the project such as the originality and the credibility of the researchers and prepares a procurement package (4.1.3). The procurement package is also referred to as an acquisition package since AIRMICS is acquiring the services of the Contractor. The procurement package consists of the keywords used and the result of the DTIC search, the PR & C (Purchase Requisition and Commitment) form, the Work Unit Summary Form, the technical evaluation, and an approval letter from the local management. As stated, the PR & C form contains the names and addresses of the principal parties to the Contractor. In addition, it also contains the contract. The next step is to send the procurement package (4.1.3) to the local management such as the division chiefs and the AIRMICS director for final approval. It is then sent to the contract/finance office also for evaluation and approval. The output from this level is a completed procurement package (also referred to as the contract).

Level 4.2 - Awarding the contract: This process consists of collecting signatures from all parties related to the contract: the director of AIRMICS, the contract/finance office, the COR and the Contractor. After the contract is awarded, the COR enters the proposal and the work/project description into the DTIC database. Since this is not decomposed further, it is shown only at the 4.0 level (*Figure 5*).

Level 4.3 - Monitoring the contract: Monitoring the contract consists of three processes, planning for review (4.3.1), conducting the In Process Review (4.3.2) and changing/updating the contract status (4.3.3). This is shown in *Figure 7*. After the contract is awarded, the COR schedules the review based on the project review dates given by the Contractor in the proposal. The process, "planning for review" (4.3.1) communicates with the external entities: Contractor for setting up and confirming the review schedule; the local management, consisting of the COR's

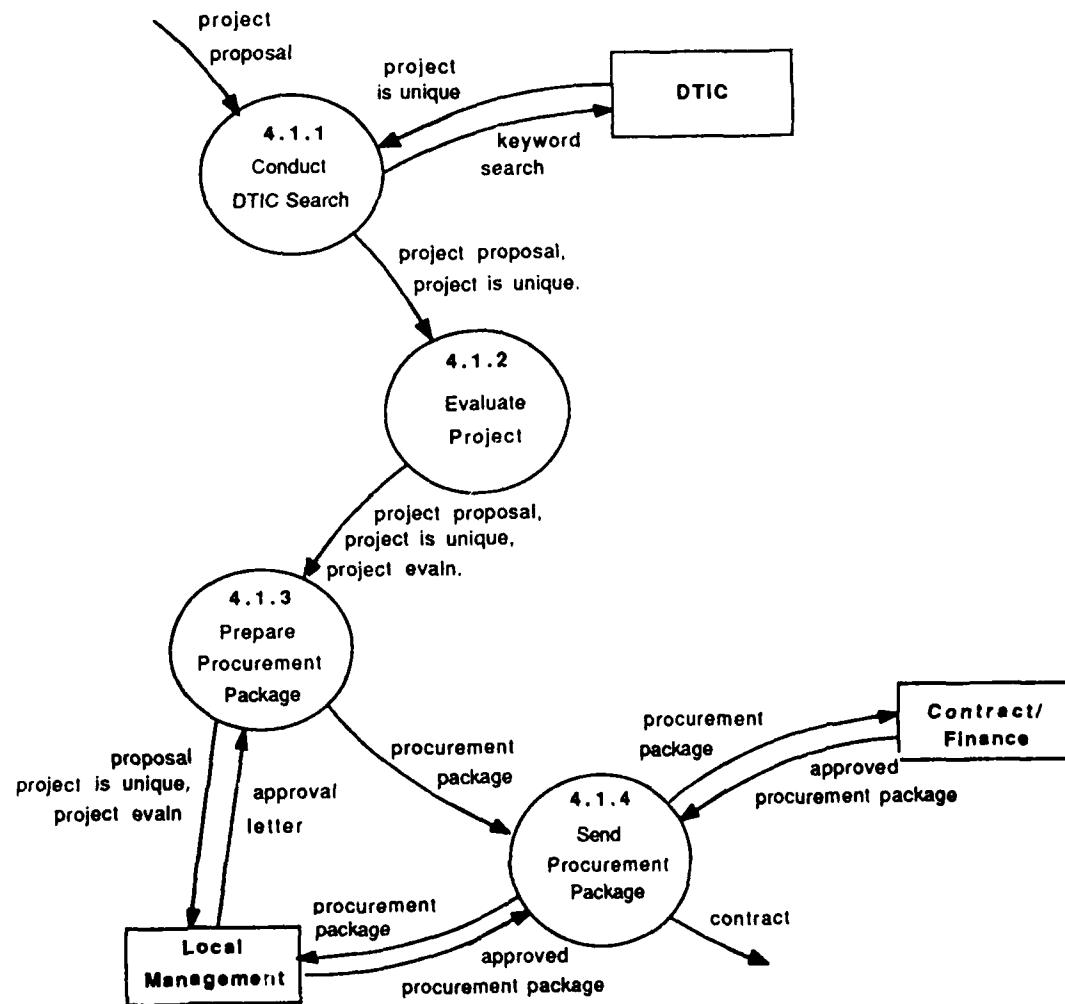


Figure 6: Preparing the Acquisition Package

division chief, for approving the TDY form; and to the entity contract/finance office for financial processing related to the TDY and AIRMICS funds. This process (4.3.1) updates the TDY file for assigning the expenses to the projects. The next process in monitoring the contracts is the IPR (4.3.2), that communicates with the Contractor regarding contract changes and contract progress, which are communicated to the process "change/update contract status" (4.3.3). This process communicates the changes to the contract/finance office to obtain approval of changes to the contract if necessary and communicates them to the Contractor. The progress/changes to the contract is also updated in the data store, "Task file."

Level 4.4 - Completing the contract: This process consists of conducting the final review (4.4.1), releasing the resources (4.4.2) and closing the contract (4.4.3). This is illustrated in *Figure 8*. The final review is triggered by the order to close the contract from the process "monitor contract" (4.3). The process "conduct the final review" (4.4.1) receives the final report from the Contractor. This report is sent to the contract/finance office with a statement that the work was accomplished satisfactorily. The COR updates the final report in the DTIC database as well as the task file which is within AIRMICS. The 1498 shown in the diagram is a special form used with the final update to DTIC. The process, "Release Resources" (4.4.2), frees up the resources, namely, the CORs, the researchers and equipment, and this information is updated in the store, "Resources." The next process is to close the contract (4.4.3), which involves the COR completing certain administrative tasks internal to AIRMICS.

3.1.6 Conclusion

AIRMICS would benefit considerably from the IOIS tools as follows: (1) The GDSS tools could effectively augment the process of identifying the areas of interest, selecting the projects to be funded, and developing a research plan. Since selecting and using the tools requires considerable experience, AIRMICS would benefit from a session planner that provides assistance for determining attributes such as the duration of the session, the specific tools to be used, etc. (2) The RME tools could be used for managing the resources, which in this case are funds, manpower, equipment and time. RME would support the processes of estimating budgets for projects which could feed into the planning process, managing the deadlines, assigning costs to specific projects, making personnel assignments, tracking personnel and equipment assignments, assisting in filling

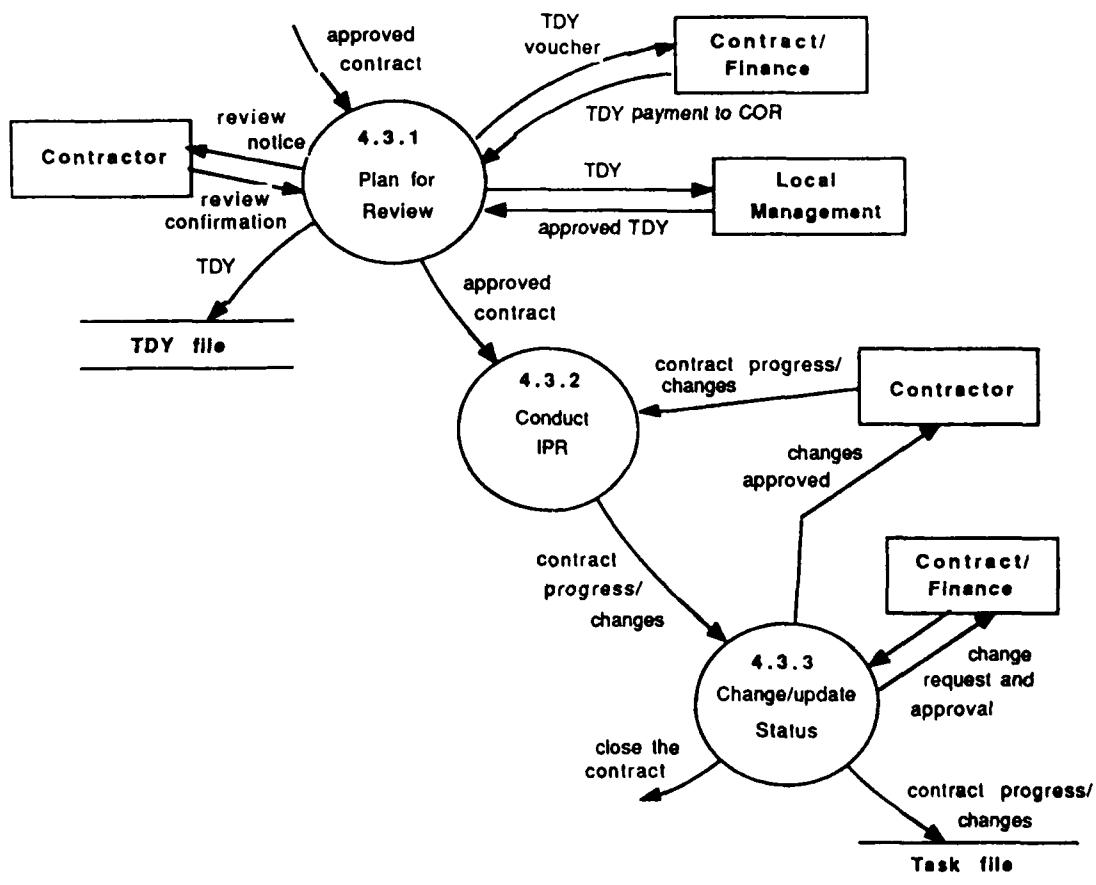


Figure 7: Monitoring the contract

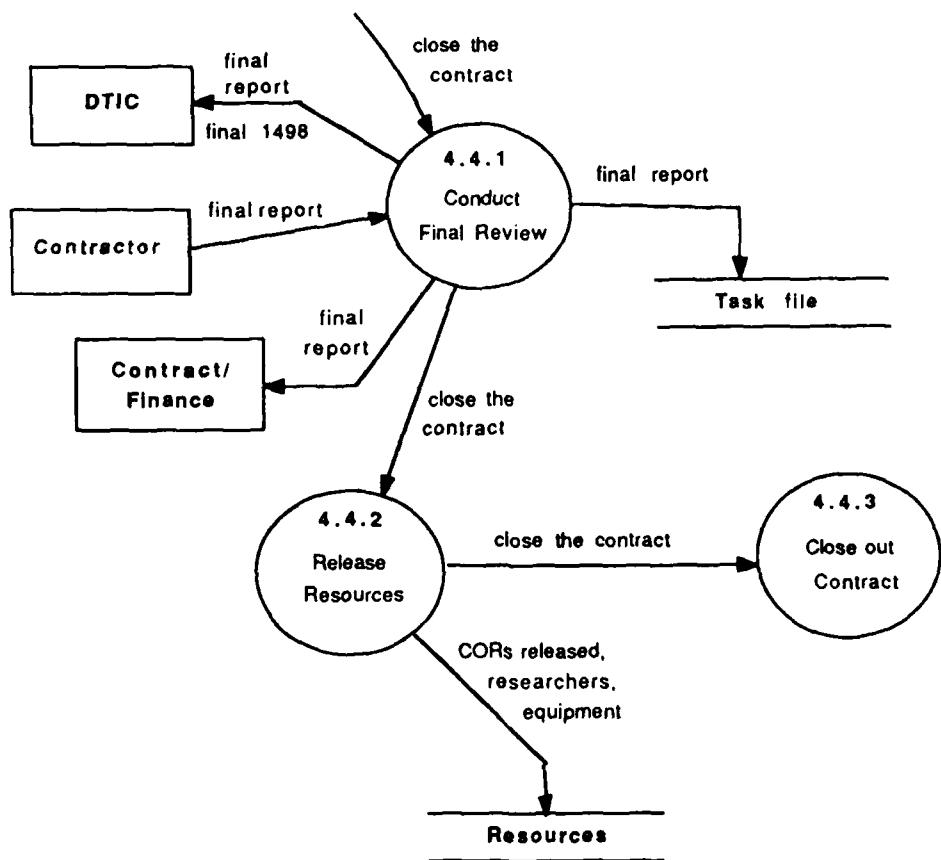


Figure 8: Contract Completion

Table 3: Tools to Support AIRMICS functions

Functions	Tools	Comments
1. Deciding on projects to fund	GDSS tools	
2. Developing RDTE plan	GDSS tools	Research expenditures
3. Estimating the budget	RME + DB	From project histories
4. Estimating training requirements	RME + DB + KB	
5. Estimating equipment requirements	RME + DB + KB	
6. Recalling project histories	RME + KB + DB	For estimating budgets
7. Scheduling meetings	Scheduler + KB	
8. Scheduling resources	Scheduler + RME	Such as meeting rooms.
9. Monitoring deadlines	RME + Scheduler + KB	
10. Alerting messages	AIMAIL + KB	Such as deadlines
11. Finding Keywords	RME + KB	For DTIC and for histories
12. Filling TDY forms	RME + DB	
13. Updating calendar	Scheduler + KB	Done after TDY

out TDY forms, and updating calendars. (3) AIRMICS would also benefit from project histories that are maintained in a summarized and machine-readable format. Histories of funds, manpower, equipment and temporal requirements of past projects would aid in estimating budgets and forecasting requirements of current and future projects. (4) The DTIC search may be supported with an expert system that solicits the description of the project from the user and finds the keywords from an online version of the keyword reference manual. This description of the project can also be used to retrieve the project histories referred to earlier. A tool to assist in the Online Editor would be helpful by taking advantage of the keywords identified by the keyword help.

3.2 Hardware/Software Selection

AIRMICS currently uses a mix of IBM compatible PCs, Sun 3/50s, and Sun 386is networked together with a Sun server. The Sun workstations use the UNIX operating system, while the PCs use DOS. The IOIS architecture will work across both DOS and UNIX environments.

Arrangements were made to allow the IOIS team to use two AT&T 6386 workstations for prototype development. To date, the 6386s have been used with DOS as the underlying operating system. These machines have been acceptable; however, AT&T's use of a non-standard color graphics display has caused some difficulty when used with the Nexpert software development package. In particular, some of the input screens used when doing development work with Nexpert had their upper portions truncated. This was later resolved by using AT&T's lower resolution EGA video mode.

AT&T's System Five version of UNIX is available for these machines, and beginning in the next six month period the IOIS will be ported to the UNIX environment. The workstations are equipped with AT&T's Starlan which may be used later when testing distributed versions of the IOIS.

Nexpert Object was chosen as the development environment based upon its wealth of inferencing strategies, its database links, and its object-oriented nature. Nexpert appeared more attractive than its competitors due to its advertised ease of integration with other programming languages (in particular C). This has not proven to be the case as yet. Since DOS was chosen as the initial operating system, a PC (DOS) version of Nexpert was purchased. It was discovered that many of the features that appeared attractive were not yet implemented in the PC package. On January 11, 1989, an upgraded version of Nexpert (1.1) was delivered. It appears to be substantially improved over the earlier version, but time has been lost in overcoming the deficiencies of the earlier version, and more time will be spent in upgrading the development from the previous to the current version.

Work is also proceeding in the C programming language. C can be integrated more easily than other languages with Nexpert, and it is anticipated that much of the prototype will ultimately be coded in C. C should also make the transition to the C-based UNIX environment easier.

3.3 Nexpert Review

3.3.1 Why Nexpert?

As part of the IOIS project, software development environments were evaluated. Since expert system principles were going to be incorporated into the IOIS prototype, only environments with these principles were considered. Other important features included portability, hardware requirements, integration with other products, and expert system specific features. DOS was to be used for this prototype, but the ability to port applications to UNIX at a later time was a prime factor.

Five well-known expert system shells were chosen for detailed study after an extensive survey of available expert system shells including EXSYS, KEE, GOLDWORKS, and TI PERSONAL CONSULTANT PLUS. Other packages were eliminated for a variety of reasons presented in the September hardware/software selection report. Neuron Data's Nexpert Object was chosen because it had all of the features advertised by the others and few of the faults.

3.3.2 Product Support ✓

State-of-the-art software is not always as good as its marketing suggests. One roadblock is poor documentation: Nexpert's manuals often omitted relevant examples and listed incorrect syntax. Many times we wondered if we were making a programming error, if the code example we were using was incorrect, or if what we were trying to do was not yet implemented in the current version of Nexpert. Calls to Nexpert's telephone help line often resulted in speaking to people who were unable to understand the problem or unable to communicate the answer.

3.3.3 Learning Nexpert

We discovered the best way to learn how to use Nexpert is to attend a school sponsored by various companies affiliated with Neuron Data. An IOIS team member went to a Callable Interface school sponsored by Bechtel Corporation. The cost of this school (over \$1,400) was money well spent, ✓ however, because the Callable Interface became a useful tool. Before attending, several months were spent trying to merely compile and link a Nexpert application. Compiling and linking

a Callable Interface application, although complex,² proved to be necessary for creating the current prototype and is likely to be even more prominent in future prototype versions.

3.3.4 Nexpert's Features

Nexpert is advertised as being "object oriented," yet it falls short of the true definition of object oriented. All variables in Nexpert are global, making it impossible to encapsulate data and methods and use messages as the only means of communication between objects. In its defense, however, Nexpert easily allows the creation of elaborate class and object structures with flexible inheritance options.

- **User Interface**

Nexpert has a wealth of features, but some of them are extremely difficult to access and use. For example, Nexpert has a state-of-the-art graphic development interface (built around Microsoft Windows PC version). Unfortunately, a professional-looking interface for an application cannot be created in the Development Environment. Such an interface requires the use of either Nexpert's Callable Interface and writing menus in C or using NORT (Nexpert Object Run Time). Either of these options force the developer to adopt a more conventional programming style.

- **NORT**

NORT has the advantage of not forcing developers to program in C and does not use MicroSoft Windows since it is text based. Its capabilities are extremely limited, however. For example, overlapping windows with menu choices are not possible; and every input screen built with NORT forces at least three presses on the return key before exiting it (press Y or N to validate screen, etc.). Display of output from NORT is also very weak. A simple IF - ELSE control statement is available in .txt files, but little else can be done to manipulate the data.

Within the case of NORT, a file with an extension of .rtd must be created to control the initialization of the program. Files with the extension of .frm are used to create menus

²involving numerous compiler options, linking approximately 130 .obj files and two libraries, and the use of overlays

and other types of screens, and files with a .txt extension are used for displaying data. Although a high level "paint" type tool exists to aid in creating .frm files, in most cases, a text editor must be used to manually create .frm files. A programming style of syntax is imposed on all three types of files.

- **Callable Interface and C**

Nexpert is organized with its AI library consisting of a core around which other modules are attached. For the PC, the development environment was created using Microsoft Windows, and is a separate module that passes messages back and forth from the AI library. Text or graphics based applications can be created by stripping off the Windows-based development environment and making calls to the AI library through the Callable Interface.

The Callable Interface corrects most of the problems of NORT and the Development Environment, yet adds new problems of its own. Programmers must be proficient with C and must deal with the compile and link times. If these hurdles can be overcome, however, Nexpert begins to show some real potential. Every function available within the Development Environment is also available through the Callable Interface; the opposite is not true.

Of most importance to "real world" applications is the ability to create innovative, professional interfaces. In some cases, the Development Environment can be used to create a KB, and the Callable Interface can be used to create an attractive interface. If a programmer is proficient with C and has a good windowing library, this is a fairly easy task and takes no more time than would be spent creating a NORT application. If some of the functions needed in the knowledge base are not supported in the Development Environment, the Callable Interface can be used for this as well.

An important class of functions in Nexpert are only available when using the Callable Interface. These allow a user to investigate the relations and values of classes, objects, and properties within a Nexpert KB. They also allow relations and values of classes, objects, and properties to be easily changed. For example, within the Development Environment a program could be written to allow a user to determine if "duck" belonged to the class of "animals." Trying to change any of the property values of "duck" (color, feathers, diet, etc.) would be nearly impossible from the Development Environment; from the callable

interface, it would be easy. Other information that relates to "duck" might be at what level in a tree structured hierarchy it was found: Is it directly linked to "animal" or are there intermediate objects between birds, or whatever? Again, with the Development Environment this is difficult, but with the callable interface it is easy.

Real time applications can be developed through the Callable Interface by using the Polling function (not available in the Development Environment). This periodically polls to see if external inputs have been sent to the Nexpert application. While this was not essential to the IOIS project, it is an extremely powerful feature not usually found in expert system shells.

- **Future Neuron Data Products**

The PC version of Nexpert is currently in version 1.1; Version 1.2 is scheduled for release in June 1989. Better access to classes, objects, and properties is promised, and hopefully some of the current bugs will be corrected. Version 2.0 is underway, but no estimates on delivery dates have been set.

An interesting product currently in the beta (test) stage is Neuron Data's Client-Server module. This allows a single knowledge base to reside on a central server and lets multiple users access the knowledge base from their PCs, Macs, or workstations. The Callable Interface module resides on each machine accessing the central knowledge base. Remote procedure calls are used to pass parameters from the Callable Interface to the server. To the user, it appears that he/she has sole access to the knowledge base and it appears to reside on the user's machine.

Another product currently in the beta stage is based on the tool originally developed and used internally by Neuron Data to create the interface of the Development Environment. It has been converted to a high level graphics screen builder that can be used from within the Development Environment to create custom graphic user interfaces. Neuron Data plans to create a text based version of this tool that could also be used from within the Development Environment. When these tools become available, Nexpert will finally allow rapid prototyping.

A Nexpert software engineer noted that the current architecture of Nexpert prevented implementing true object-oriented features. He also said that work was underway to create a new product that would be object oriented and combine the best features of the current product with the full range of object oriented attributes. The new product is unlikely to be seen for two to three years, however, and will not be compatible with the current product.

3.3.5 Nexpert Pros and Cons

- **Pros**

1. Many hardware platforms are supported.
2. Access to most commercial databases are supported.
3. Good Knowledge base editor.
4. Trace features for debugging the KB.
5. Flexible inferencing.
6. Unique and useful class and object structure.
7. Linked to the C programming language.
8. New products and enhancements are in the pipeline.

- **Cons**

1. Poor documentation and support.
2. Too many "bugs".
3. Forced to use the Callable Interface too often.
4. NORT has too many limitations.
5. Poor access to objects and classes from within the development environment.
6. Idiosyncratic syntax for manipulating objects (from rules).

The cons should not be taken as an indication that the product is unacceptable or unusable. On the contrary, Nexpert is one of the most powerful tools on the market, and it is expected that the cons will diminish over time as the product matures.

Nexpert has a steep learning curve. Even after becoming proficient with Nexpert, applications cannot be rapidly produced. Future enhancements that allow professional quality interfaces to be created without leaving the Development Environment will speed up application development.

4 Design

The objective of the design phase is to develop specifications for realizing the IOIS architecture in a specific office setting. The design was developed both from the case study and according to the capabilities of the available tools. The design tasks were split according to the components of IOIS: AGDSS, RME, KB/DB, Interface, Calendar/Scheduler..

4.1 AGDSS

The Expert Session Planner (ESP) is designed specifically to assist electronic meeting coordinators in pre-session planning of group meetings. ESP can assist the meeting coordinator in the selection of appropriate tools and participants for the session. In addition, ESP can scan the calendars of the session participants to determine convenient times for having face-to-face meetings. ESP thus allows planning and coordinating of electronic meetings with minimal knowledge of the GDSS tools and the processes involved in pre-session facilitation. The following subsections will discuss knowledge acquisition, knowledge representation, and the inferencing strategy employed in the design of ESP.

4.1.1 Knowledge Acquisition

Knowledge acquisition is perhaps the most difficult and tedious phase of any ES project. Effective knowledge acquisition strategies must be employed for the ultimate success of the system. Ideally, designers (or knowledge engineers) should already be familiar with the basics of the problem at hand; if not, they should review applicable textbooks and references before approaching the human expert. Only by establishing a mental model framework will the designers be able to file the anecdotes and rules-of-thumb acquired from interviewing the expert. Also, a minimum set of knowledge in any area is a prerequisite for posing incisive questions.

For the prototype ESP, the authors met over a period of three months with a panel of local GDSS experts to determine the factors involved in the tool selection process. This panel of experts

included group facilitators with several years of GDSS session experience, and tool developers and researchers. IOIS team members had moderate experience with GDSS terminology and methodology through prior study. In the final stages of knowledge acquisition, a GDSS session involving issue analysis and organization was conducted at the University of Arizona to categorize the factors involved. The team members also met with Major Ted Hengst from AIRMICS to determine the tools and participants needed for meetings based on job responsibilities, work interests, and departmental affiliations. This knowledge was incorporated into the system and is discussed in more detail in the implementation section.

4.1.2 Knowledge Representation

Once the underlying knowledge of the expert task has been acquired, a theoretical representation of the knowledge must be developed. Some representations include: IF-THEN rules (perhaps the simplest and most prevalent technique), Frames and Objects (popular for their powerful inheritance and hierarchical properties), and Semantic Networks (noted for their flexible, yet powerful structures). The choice of knowledge representation is perhaps the guiding force of the ultimate implementation.

An expert system for pre-session planning will benefit from a Frame and Object representation since group and tool characteristics often have hierarchical structures. For example, group participants can be classified by organizational affiliation, interests, or responsibilities. An ES which selects appropriate group participants can rely on certain inherited features instead of features which are expressed for each group member. Likewise, GDSS tools can be classified by their applicability for a particular group or problem. Inheritance of these characteristics will allow the ES to proceed more efficiently in determining which GDSS tools to select.

The initial prototype of ESP was designed with the procedures of selecting the tool and participants represented in IF-THEN rules, with the facts about the tools and participants stored in a database. When the rules are fired, the facts associated with the objects in the rules are instantiated from the database. This reduces the number of rules required in both selecting the tools and participants. The knowledge base can be further refined with the use of powerful representation techniques such as semantic nets, frames, and object-oriented approaches, to be investigated in future.

4.1.3 Inferencing Strategy

The expert task must also be analyzed as to the fundamental nature of the problem: one of design, diagnosis, or both. Design problems typically feature a relatively limited set of underlying conditions and a much greater number of goals; diagnosis problems include relatively few goals with many conditions which must be verified. For example, the problem of group selection is one of design since the number of potential participants is likely to be much greater than the number of criteria involved in the group configuration. Conversely, tool selection is primarily a matter of diagnosis; there are a relatively limited number of tools while there are many more factors involved in their selection. A forward-chaining inferencing strategy is most efficient with design problems, while a backward-chaining strategy is most economical with diagnosis tasks. Some problems may involve diagnosis and design, indicating a need for a hybrid inferencing strategy.

ESP uses an exhaustive, backward-chaining inferencing strategy with provisions for both user uncertainty (the user isn't sure of a fact he is providing) and rule uncertainty (to resolve conflicting rules).

4.2 RME

Detailed descriptions of RME functions will be described. The overall IOIS architecture (*Figure 9*) is included to show how RME interacts with the other components.

The Resource Management Expert (RME) provides assistance to AIRMICS personnel for managing resources: funds/expenses, personnel, equipment, schedules. The design of RME is based on the data flow diagrams presented in the case study. The overall process that AIRMICS goes through to plan, execute and evaluate research in the areas of interest consists of selecting tasks to be funded, collecting the budget information, allocating resources, and administering contracts. These functions have been regrouped into planning for projects, resource management and contract administration components for the design of RME.

The architecture of RME is illustrated in *Figure 10*. It consists of five major components: planning, resource management, administer contract, and database/knowledge base. Although the database and knowledge base are external to RME in the IOIS architecture, it will be treated as part of RME in order to describe its contents. In the design, all manual processes have been considered to have the potential for automation. This includes filling forms, allocating resources,

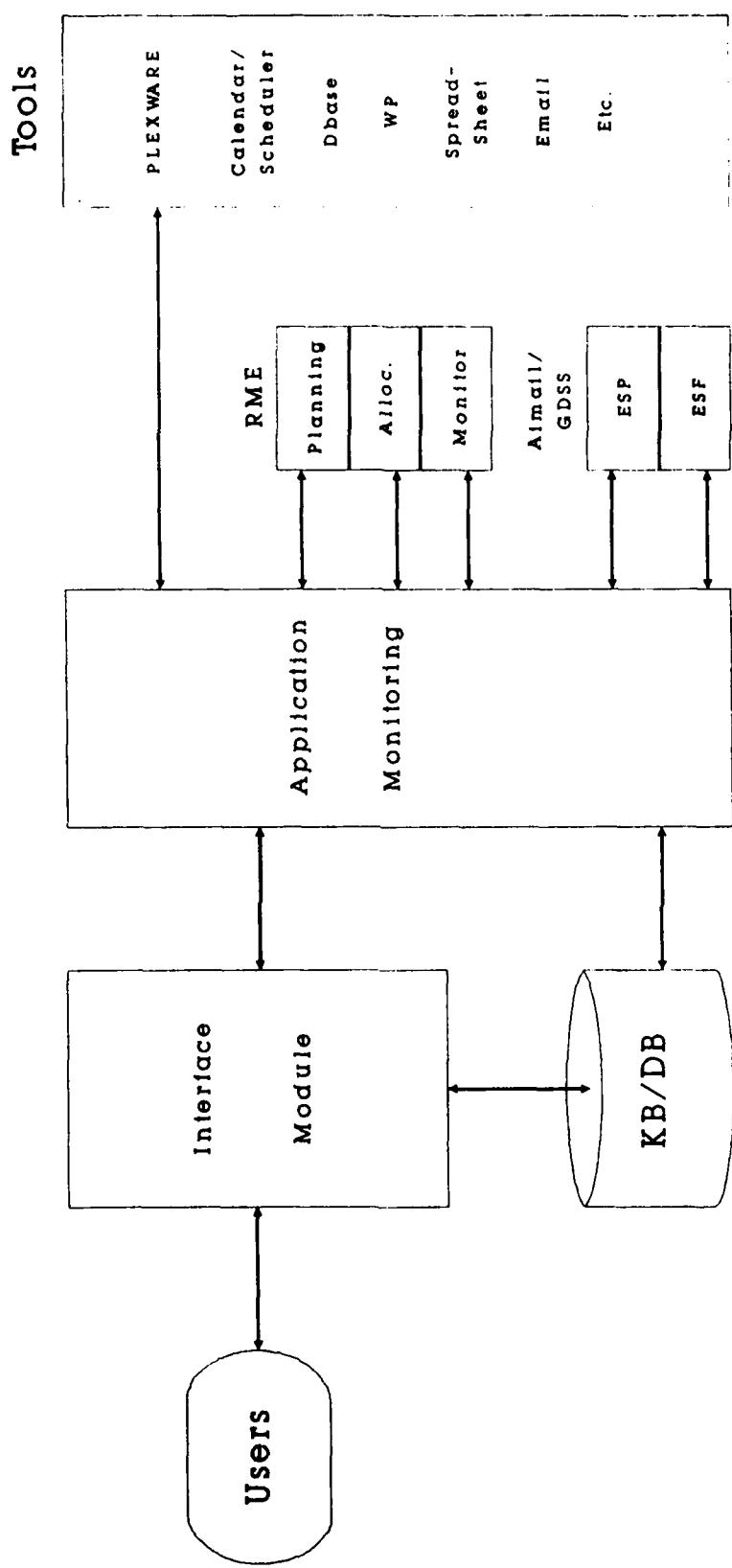


Figure 9: Architecture of IOIS

searching through databases, etc. Apart from the reorganization of the functions for the design, there are several additional changes from the manual process; first, the DTIC search is now conducted during the planning stage to avoid including projects in the RDTE plan that have already been accomplished in another branch of the DOI; second, the keyword identification is assisted with an object structure using a parent-child relationships in the interest areas; third, TDY, equipment, personnel, printing etc. costs are now assigned to individual projects. (Currently AIRMICS includes these expenses as a percentage overhead for each project, and it would benefit from better estimates and management of project costs.) Fourth, the project activity schedule, the review schedule, and the TDY schedule, are linked with a calendar. The calendar itself is regarded as a generic IOIS tool, so it is not included in the RME. The purpose of linking RME with the calendar is to provide support for generating review notices, TDY requests and approvals, and reminders for deadlines. Project management is therefore developed with monitoring schedules/activities and resources as a basis. Fifth, project histories are stored in the IOIS database for rapid access to textual descriptions, interest areas and resource utilization. Although project histories are currently stored in the DTIC database, the management of AIRMICS would benefit from rapid and ready access to the salient details of historical projects. The project histories aid in planning and estimating project resources.

The components are briefly described here.

Planning: The function of this component is to support the planning executed by the GDSS module by providing information on resource availability, by providing access to historical information on projects and by providing resource estimates for projects on the proposed task list. Access to historical information is assisted with the object oriented keyword help facility. The results of the planning phase are updated in the IOIS database using standard forms for inputting variables such as whether or not a project is approved, the level of funding and so on.

Resource Management: This module is intended as a utility to provide resource management functionality to other components of RME as well as for IOIS. It includes subroutines for estimating resources, for obtaining a status on the resources from the database, resource allocation, resource update, resource monitoring, and resource usage . Resource allocation includes allocation of funds (which were already earmarked in the planning process), personnel and equipment. It is accomplished intelligently with the resource profile stored in the database, the domain

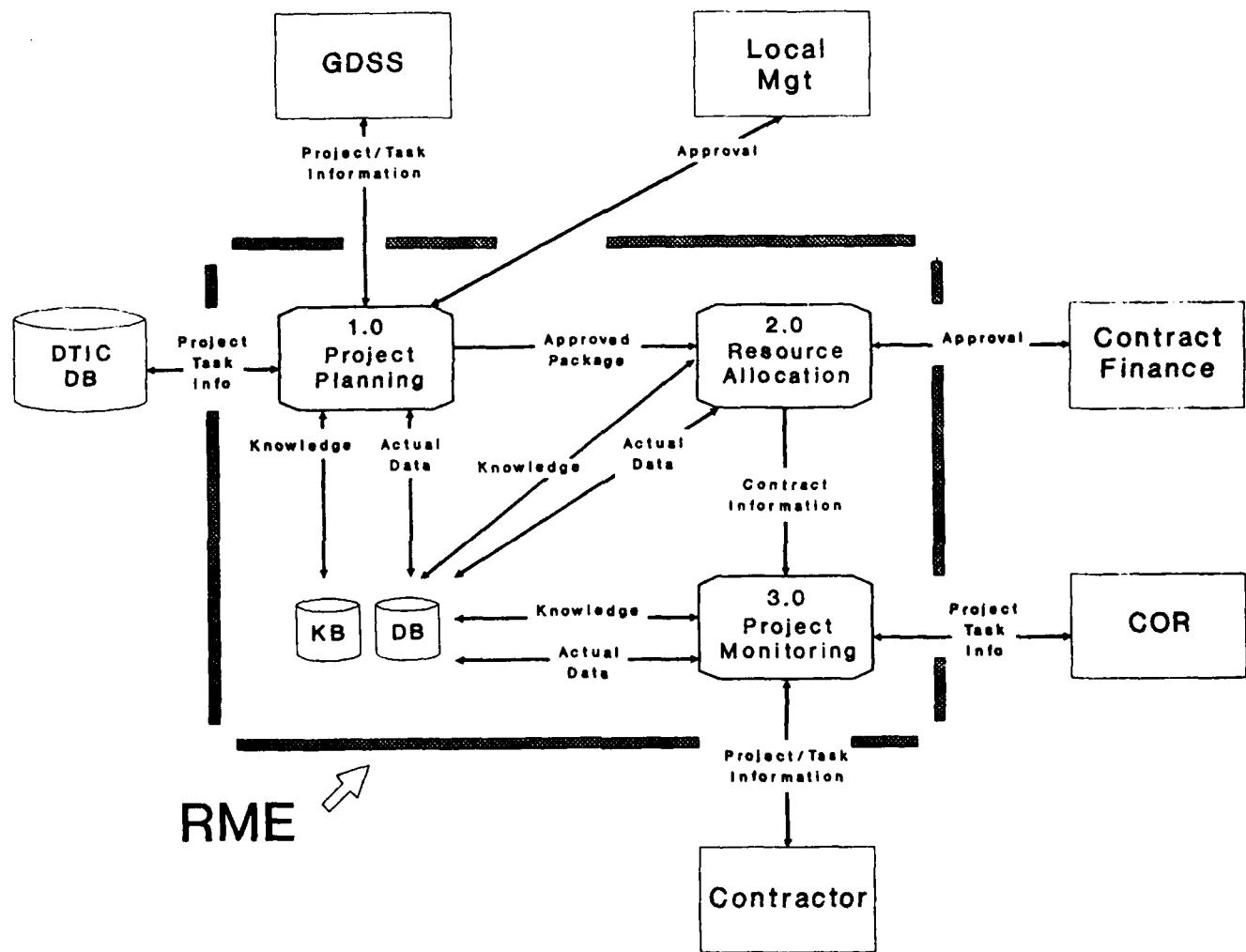


Figure 10: Architecture of RME

knowledge of the AIRMICS interest areas stored in the knowledge base, and the allocation rules also stored in the knowledge base. The module provides a report if necessary on the status/usage of resources for an individual project.

Administer Contract: Project management involves initiating projects, monitoring schedules, monitoring resources and terminating the project. Initiating a project consists of preparing the acquisition package, awarding the contract, and committing the resources. After a project is initiated, the major activity of AIRMICS from the perspective of RME is to monitor the project. Although the actual tasks are executed by the COR, RME provides the necessary support. By monitoring the schedule for review dates, completion of project activities, scheduling of reviews and monitoring of resources. This module also has a subroutine for carrying out the contract closing procedures (updating the DTIC database, etc.).

KB/DB: The RME KB/DB component is accessed by the three modules. The database contains records of entities that are relevant to RDTE, such as projects, personnel, equipment, etc.; the knowledge base is required for various reasons, such as retrieving the list of products delivered for an earlier project, estimating budget for a task based on the description of the task and the project histories stored in the database, allocating personnel based on their skills and the project requirements etc. This component will be described in detail later.

Figure 9 shows that RME has five external interfaces: the GDSS module, the local management, the DTIC database, the contractor, and the contract/finance office in Fort Mc Pherson. The GDSS component is active in the planning phase. The local management is regarded as being external to RME. RME interacts with it to obtain approvals for the acquisition package, TDY, etc. Since all projects are checked for uniqueness with the DTIC database, this external entity is also included in the RME design. As mentioned in the case study, the contract/finance office has financial responsibility for the project. RME therefore has interaction with this entity for all AIRMICS activities involving finances.

A more detailed description of the design for RME is now presented. As stated earlier, RME consists of three components which are planning for projects, management of resources, and administration of contracts. The third component is a utility module that provides resource management functionality to the other two components. The rationale for isolating the utilities is to factor out the generic resource management functions into a separate module as a first step

towards generalizing the design for other army offices.

1. Project Planning

The detailed view of the project planning component is shown in *Figure 11*. The planning component assists AIRMICS personnel in planning and budgeting for projects. A group decision support environment is envisioned for supporting the planning. RME assists this phase by providing information on past projects from the database/knowledge base and by obtaining and providing information on resource availability. It consists of five major components: 1) Retrieve project histories, 2) Estimate resources, 3) Resource status, 4) Resource monitoring and 5) finalizing the RDTE plan.

Retrieve Project Histories:

Project histories are essentially accumulations of the RDTE experience, that may be used as a basis for making decisions on current and future projects. This component is especially relevant to AIRMICS, because one of its functions is to accumulate technical knowledge and to provide these as inputs to other branches within the DOD. The project histories include technical problems, products delivered, project costs, equipment requirements, project schedules etc. A variety of search conditions are provided for, such as retrieval by activity, retrieval by project or retrieval of history by contractor's name.

The histories are stored and retrieved on the basis of keywords which describe the project. For a current project, the keywords are identified during the planning/acquisition phase of RDTE and the data is stored/updated as the project proceeds. The information becomes historical when the project is completed. An object based knowledge structure providing a detailed classification of the interest areas of AIRMICS assists in identifying the keywords to describe a project. For the RME component, examples of these classifications are the specific interest area (OA), any sub-area (Financial application), the research approach used (Prototyping), etc. The keywords that are identified with this structure for a current project can also be used to search the DTIC database. Keyword identification is currently performed manually with the help of a keyword reference manual. Keyword help in RME is equivalent to an intelligent dictionary/thesaurus to be used to obtain the keywords to search the databases.

Resource Estimation:

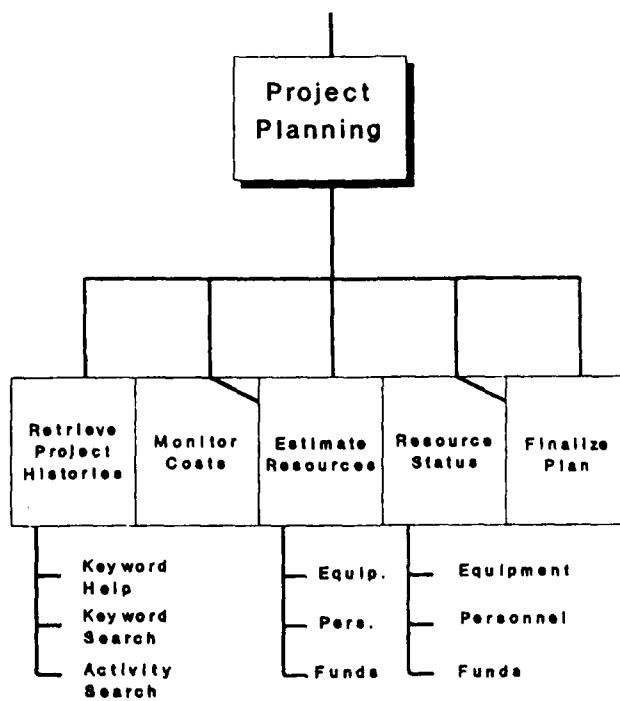


Figure 11: The Project Planning Component of RME

This module estimates the resource requirements for projects on a task list. Costs, equipment and manpower requirements are estimated from the historical information to provide rough estimates for planning purposes. Project histories are retrieved by keywords as discussed earlier. In the planning phase, RME may be called from the GDSS environment to provide either descriptive project histories or resource estimates. Each project in the task list is associated with a set of activities given as input and characterized by descriptive keywords. Given the project, the keywords, and the activity list, the resource estimation module searches through the database, guided by the knowledge structure, to identify projects similar to the given project. Based on the resources used in similar projects and heuristics used for scaling the estimates upward or downward, the estimates/ requirements are presented to the user as suggestions for resource requirements.

Resource Status:

This module serves to provide status of the resources available in AIRMICS to anyone involved in the project planning or monitoring phase. For instance, a division chief may use the module for identifying researchers who are not on any projects. The module can be invoked by another component such as the AGDSS while developing a research plan. It communicates with the external entity, the contract/finance office, to obtain funding information. Details such as the fund types and fund expiration dates are obtained. The module obtains information on the available personnel, their areas of expertise and the equipment availability from the database.

Monitoring Resources:

This module is part of resource management utility, and monitors the resources that were consumed during the planning process, such as GDSS equipment usage, personnel time, photocopying expenses etc.

Finalizing the RDTE plan:

Once a research plan is finalized, the IOIS database is updated with respect to the details of the contractor etc.

2. Resource Management

As stated earlier, the Resource Management component of RME provides resource related

utilities dealing with resources, to the other components. It can estimate resources, provide the status of all resources, allocate resources, update resources as they are utilized, monitor the calendar for any due dates, and provide a resource usage report for any project. The structure chart for the Resource Management component is illustrated in *figure 12*. The components "Resource Estimation" and "Resource Status" were discussed in *1*. The remaining utilities are: 1) Resource Allocation 2) Monitor/Update Resources and 3) Resource Usage. They are described below.

Resource Allocation:

Resources are allocated after a project is approved. As noted in the case study, the CORs and the researchers for internal projects are selected by the AIRMICS director and the division chiefs. Given the task list, RME attempts to arrive at a preliminary assignment of responsibilities for each task based on the allocation rules stored in the knowledge base and the resource profile extracted from the database. The final decision rests with the local management. For external projects, this module is concerned mostly with funds, CORs and equipment. CORs are assigned based on availability, areas of expertise and the requirements of the project.

Monitor/Update Resources

An important function of RME is to monitor and update the resources as they are consumed.

Personnel costs are estimated from length of the assignment (researcher or COR), and the compensation rate. Other costs such as printing and publications costs are recorded as they occur.

Equipment purchases are recorded in the database and any equipment usage times are captured through the interface. For instance, if a two hour GDSS session were scheduled for a project, RME is able to estimate the total cost based on hourly costs and the total scheduled time, available from the calendar/scheduler. Similarly, cost information is captured from a DTIC session based on the duration and cost per minute.

Although not shown in *figure 10*, RME makes use of a calendar/scheduler to follow activities and to generate reminders when necessary. The schedule of project activities are stored in the calendar/scheduler as soon as a contract is awarded.

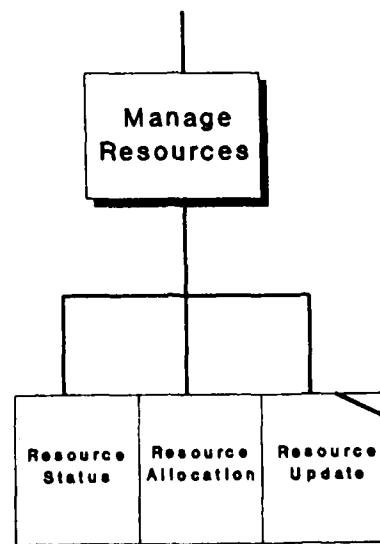


Figure 12: The Resource Management Component of RME

This module complements the function of the component, "co-ordinate reviews" under "Administer Contracts". It identifies the review dates and monitors the database regarding the final review schedule. When the review dates have been finalized, the COR initiates the TDY processing and these costs are recorded. "Monitor resources" also has an additional function, which is to update the changes in project or delivery schedules that may occur as the project progresses. The updates are carried out in the database as well as in the calendar/scheduler using standard input forms.

Resource Usage Reports:

Resource usage reports are a management tool for assessing costs of individual projects, and can be used for accounting purposes as well. Equipment purchase, equipment usage, personnel time, printing and publication, TDY, and training are all costs to be assigned to individual projects. There are some overhead costs also, such as the GDSS tool usage costs. The reports show the costs for each project and the overhead costs.

3. Administer Contracts

Administering contracts consists of initiating the projects, monitoring them, and closing the contract, as shown in *Figure 13*. RME provides support in each of these phases. Initiating a project consists of preparing the acquisition package, finalizing the package, awarding the contract and allocating the resources. During the preparation of the acquisition package, a keyword help facility assists with the DTIC process, and electronic support is assumed for routing the PR & C documents. In Monitoring projects, RME assists with resource monitoring as well as activity monitoring. Any costs are recorded as described earlier. When a contract is completed, the IOIS and DTIC databases are updated, resources are freed, and resource usage reports generated.

Initiate Projects:

In "Initiate Projects", the acquisition package is processed and resources are committed.

The acquisition package consists of a statement of the result of the keyword search, the technical evaluation of the project, the summary of the work (PR & C form) and the approval letter. As described earlier, keyword help and automated support for accessing the DTIC database are provided for assisting with the preparation. Again, a forms interface

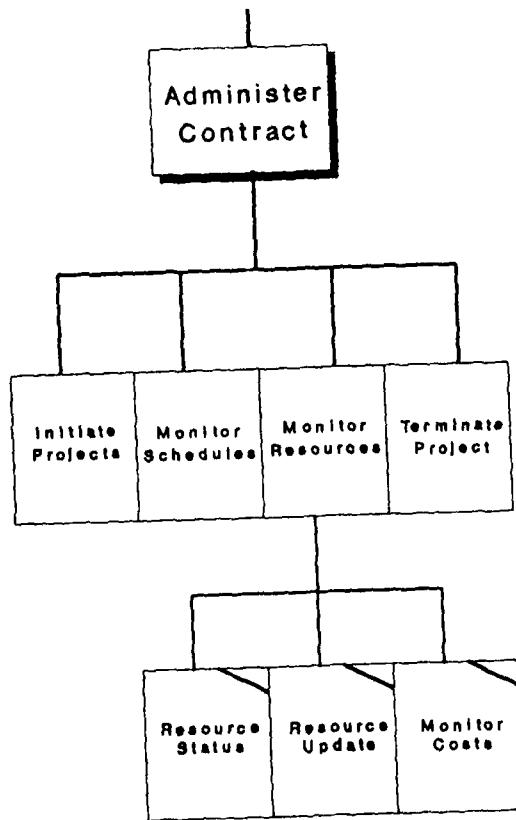


Figure 13: The Contract Administration component of RME

is assumed for filling out the PR & C and the Project Technical Evaluation. Finalizing the acquisition package consists of obtaining the approval from the management. This is performed for each project that was on the approved project list from the GDSS component.

After the signatures are collected and the contract is awarded, this module initiates a number of events including updating the project status to "active" in the database, and initiating the monitoring of activities/schedules.

Monitor projects:

Project monitoring consists of co-ordinating the reviews, monitoring activities and monitoring resources. All of these components have interactions with the general resource management facilities discussed earlier.

The component "Monitor Activities" alerts the subroutine "co-ordinate reviews" when a review for a project becomes imminent. This module then initiates coordination of the review date and time with the contractor, and also initiates TDY processing.

The "Monitor Projects" module follows the progress of a project by monitoring the project activities and status of deliverables. The COR updates the schedule as the project progresses and the activities are completed. Change requests are forwarded to the contract/finance office for approval. When the project is completed, this module sends a flag to the module to "close contract".

This module also makes use of the functions in the "manage resources" component to monitor the status and usage of resources. This may involve temporarily assigning a researcher to a certain project, making arrangements to cover absences etc.

Closing the contract: When the module "monitor projects" signals the completion of the project, the component "close contract" is activated for the closing procedures. The final project report is updated in the IOIS database as well as in the DTIC database. "Close contract" also sends a final 1498 form to the DTIC database.

4.3 IOIS Database and Knowledge Base Design

4.3.1 Introduction

In IOIS, DB and KB are coupled together. Using KB as an intelligent component, DB system supports more sophisticated queries and updates, e.g., query navigation and data integrity maintenance. DB supplies facts (data) to KB so that knowledge can be instantiated. For example, if an office has different purchase order forms and order procedures for different equipment, KB system can identify an appropriate form and procedure given an equipment. However, actual order form and procedure are stored in DB. Therefore, KB system must access DB in order to instantiate its knowledge. Because of this association, development of DB and KB must be performed together. A method for DB/KB design was developed based on the following criteria:

1. **Scope of Method:** The method must facilitate the entire development cycle of DB/KB.
2. **Uniformity:** In order to maintain the consistency, the method should use a single representation scheme at least for the analysis and conceptual & logical design phase. Physical design may still rely on both hardware and software.
3. **Ease of Use:** The method should be easy to use, and the developed DB/KB should also be easy to implement. IOIS prototype will be developed by using NEXPERT (an expert system shell), so DB/KB must be implementable on this software environment.

Currently, very few methods satisfy the first two criteria. After satisfying the first two, there is only one method satisfy the third criteria at this point that is the Structured Entity Model (SEM) method.³ Our preliminary investigation of SEM and NEXPERT indicated that NEXPERT can strongly support SEM knowledge representation (frame-based semantic tree). Also SEM can produce a relational schema as a result of design phase, and NEXPERT supports dBASE III+ which is a psuedo relational-based database.

In this section, the Structured Entity Model (SEM) is first briefly described. Second, the description of database schema is provided. Then representation of knowledge using SEM is discussed at the end of this section.

³Higa, K, End-user Logical Database Design: The Structured Entity Model Approach, Ph.D. dissertation, the University of Arizona, 1988

4.3.2 Structured Entity Model (SEM)

The Structured Entity Model (SEM) is a frame-based semantic tree originating from System Entity Structure (SES). SEM represents data semantics using entities, attributes, and three types of relationships: aspect, specialization, and multiple decomposition. An entity is an event or an object about which users wish to collect and store information. In SEM, entities are represented by frames. However, there is a distinction between an entity and an entity class. Projects, employees, and equipments are examples of entity classes. Attributes are used to describe entities by providing them with descriptive properties such as name, shape, color, etc. There are two types of attributes: identifiers and descriptors. An identifier uniquely identifies an entity occurrence, and descriptors describe the state of the entity occurrence. SEM also has extended attribute types (e.g., a formula, a rule, a procedure, etc.). Relationships represent associations among entities in the real world. Semantic meaning of relationships is indicated by the connectivity between entities (one-to-one, one-to-many, and many-to-many). Participation of an entity in a connectivity may be either optional or mandatory. Relationships in SEM can be categorized as aspect relationships, specialization relationships, and multiple decompositions.

SEM is constructed by decomposition and coupling. The decomposition process generates the structure of an entity, while coupling determines how entities are coupled together. Stamp-coupling⁴ is a coupling mechanism used in SEM. After an entity has been defined in the structure, the mechanism couples later appearances of the entity with its original definition. As a result, the mechanism ensures that an entity will not be decomposed more than once. Although, coupling of any combination of entities is possible in an SEM diagram, a user can determine correctness in coupling by checking the association (meaning) of the coupled entities.

4.3.3 Description of Database Schema

Entities discussed in this subsection are by no means an exhaustive set of entities in the AIRMICS office. However, they represent typical entities of the IOIS prototype. The prototype database using a SEM diagram is depicted in Figure 14. In this subsection, entities having a non-composite key, called simple entities, were first described. Then entities having a composite key, called intersection records or complex entities, were discussed in the following subsection.

⁴Schneyer, R., *Modern Structured Programming*, Mitchell Publishing, Inc. Santa Cruz, Calif. 1984, pp. 121-122

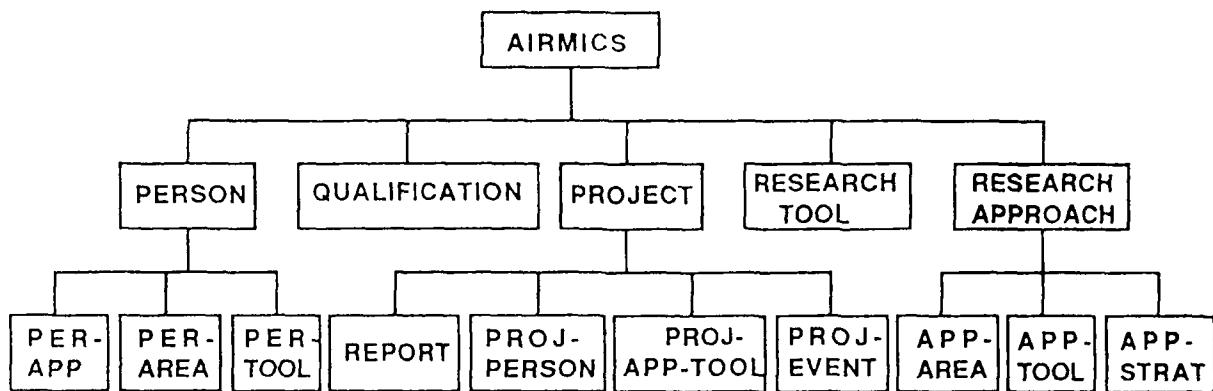


Figure 14: A Prototype IOIS Data and Knowledge Structure

1. Simple Entities

For the prototype DB, the following six simple entities were selected from aforementioned case study.

- **PERSON** entity describes AIRMICS personnel.
- **QUALIFICATION** entity contains requirements such as educational level, expertise, etc. to each research approach.
- **PROJECT** entity describes a research project. An instantiation of this entity will provide sufficient information about a research project.
- **REPORT** entity is a generalization of all report types such as **PROPOSAL**, **PROGRESS**, and **FINAL**.
- **RESEARCH TOOL** entity includes both hardware and software research tools used in AIRMICS.
- **RESEARCH APPROACH** entity describes various research approaches used in AIRMICS (e.g., case study, prototype, experiment, etc.)

2. Complex Entities

The following nine complex entities were defined during the analysis/design of the prototype IOIS database using SEM.

- **PER-APP** entity describes each person's expertise in particular research approach.
- **PER-AREA** entity contains each person's research area and experience level in the area.
- **PER-TOOL** entity describes each person's skill level to particular research tool.
- **PROJ-PERSON** entity is a result of the many-to-many relationship between a project and a personnel. Using a **PROJECT** identifier, all personnels who work at the project can be identified.
- **PROJ-APP-TOOL** entity represents the ternary relationship among a project, a research approach, and a research tool. An identifier of this entity is composite of **PROJECT**, **RESEARCH APPROACH**, and **RESEARCH TOOL** identifiers. Hence using a **PROJECT** identifier, a set of research approaches with tools can be identified.
- **PROJ-EVENT** entity represents the relationship between a project and an event. Given a **PROJECT** identifier, there will be a set of events.
- **APP-AREA** entity describes the fitness between research approach and research area in two levels, primary and secondary.
- **APP-TOOL** entity describes the fitness between research approach and research tool in two levels, primary and secondary.
- **APP-STRAT** entity describes the fitness between research approach and research strategy in two levels, primary and secondary.

The schema described above (both simple and complex entities together) represents was developed using SEM method. Therefore, the entire prototype database is consistent and has a natural association to the prototype knowledge base, which is described in the following subsection.

4.4 Knowledge Base Design Using SEM

In IOIS, there are two types of knowledge, factual and structural. Factual knowledge are instantiations of organizational events and objects, i.e., actual data. Thus they are stored in a database as described in the previous section. Structural knowledge are further categorized into static and dynamic. In this section, design of both static and dynamic structural knowledge are described.

1. Static structural knowledge

Static structural knowledge (SSK) include descriptions of organizational events and objects, collectively called entities, and associations between those entities. In fact, the final SEM diagram is identical to SSK. Thus the design of SSK is already completed when an SEM diagram is constructed for the design of a database schema. Organizational security measures and data retrieval/update policies can be incorporated into SSK. Therefore, SSK can be used as a security guard and policy enforcer of the organizational DB. The design of SSK must precede the design of dynamic structural knowledge because the latter use components of the former.

2. Dynamic structural knowledge

Any task in an office typically involves multiple entities. Those entities are organized and form a specific structure on which various inferencing necessary to accomplish a task can be performed. The entity structure, which is formed for a specific task, and inferencing patterns are called Dynamic Structural Knowledge (DSK) in IOIS. DSK is designed first by collecting required entities for the task from SSK then a specific association among entities is constructed to form a customized SEM diagram for the task. The designer then defines inference patterns on the diagram. The actual design of DSK is illustrated through an example, 'the DSK design for the keyword search function,' in the following.

DSK Design Procedure:

Step 1. Collect required entities from SSK.

Example: The objective of the keyword search function is to find appropriate project reports given a set of keywords. Keywords used in this task are related to attributes research strategy, approach, and tool. Therefore, PROJECT (contains attributes strategy and research area), RESEARCH TOOL (contains the attribute tool), RESEARCH APPROACH

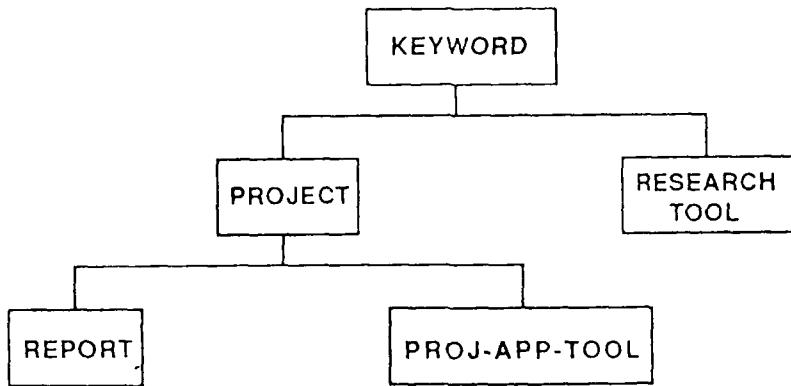


Figure 15: The SEM Diagram for Keyword Search Function.

(contains the attribute research approach), PROJ_APP_TOOL (contains ternary relationship), and REPORT (the goal) entities are selected from the SSK in Figure 14

Step 2. Construct an SEM diagram using collected entities.

Example: Any keyword belongs to PROJECT, RESEARCH TOOL, PROJ_APP_TOOL, or RESEARCH APPROACH entity thus KEYWORD entity (dummy root) initially consists of those four entities. In this particular example, the rest of structure is same as the original SSK. Therefore, the structure is also copied from the SSK. In order to simplify the SEM diagram, RESEARCH APPROACH entity is deleted because all keywords related to RESEARCH APPROACH also exist in PROJ_APP_TOOL entity. (The SEM diagram is shown in Figure 15.)

Step 3. Define inference patterns on the SEM diagram

Example: Two keywords are used as input data for the prototype implementation. The following four Inference Patterns (IP) exist in the DSK designed in Step 2 with two keywords.

Common Input: Two keywords typed by a user.

IP-1: Keywords are strategy and research_area attributes (Figure 16a).

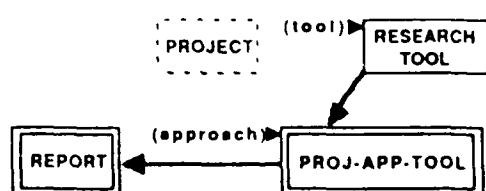
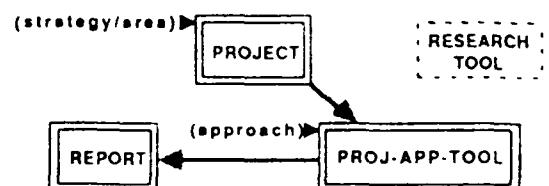
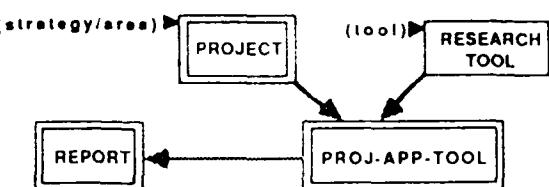
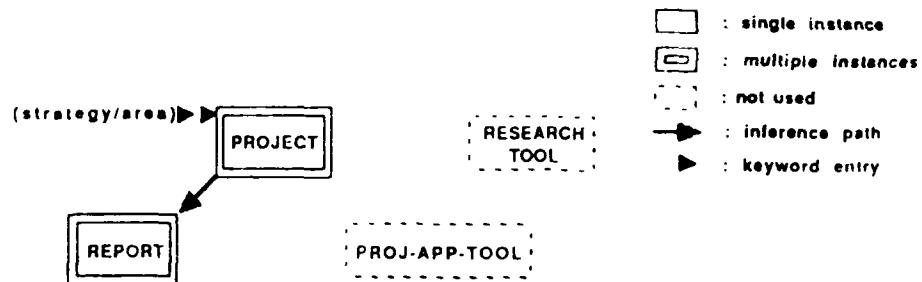


Figure 16: Inference Patterns for Keyword Search.

Given a strategy and a research area, many projects will exist. Each project has a report thus many reports will be selected.

IP-2: Keywords are strategy/research_area and research_tool attributes (Figure 16b).

Given a strategy/research_area, many projects will exist. Selected project IDs will be used to instantiate many PROJ_APP_TOOL. A research tool will identify a single tool ID which is used to finalize the instantiated set of PROJ_APP_TOOL.

IP-3: Keywords are strategy/research_area and research_approach attributes (Figure 16c).

Given a strategy/research_area, many PROJ_APP_TOOL will be instantiated (this is same as IP-2). Then a research approach is used to finalize the instantiated set of PROJ_APP_TOOL.

IP-4: Keywords are research_tool and research_approach attributes (Figure 16d).

A research tool identifies a single tool ID which is used to instantiate many PROJ_APP_TOOL.

Then a research approach is used to finalize the instantiated set of PROJ_APP_TOOL.

Output: Many REPORT which will be instantiated by project IDs from the selected set of PROJ_APP_TOOL (or PROJECT for IP-1).

Step 4. Interpret the inference patterns into Nexpert rules.

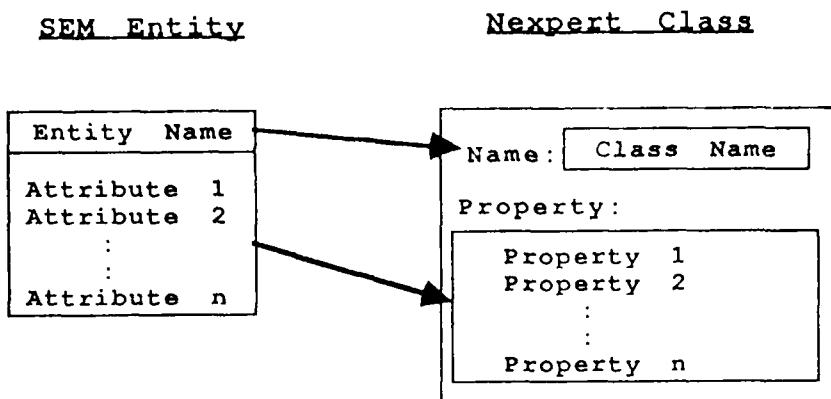
Detailed description of this step is provided in the following.

Transformation of SEM diagram to Nexpert representation.

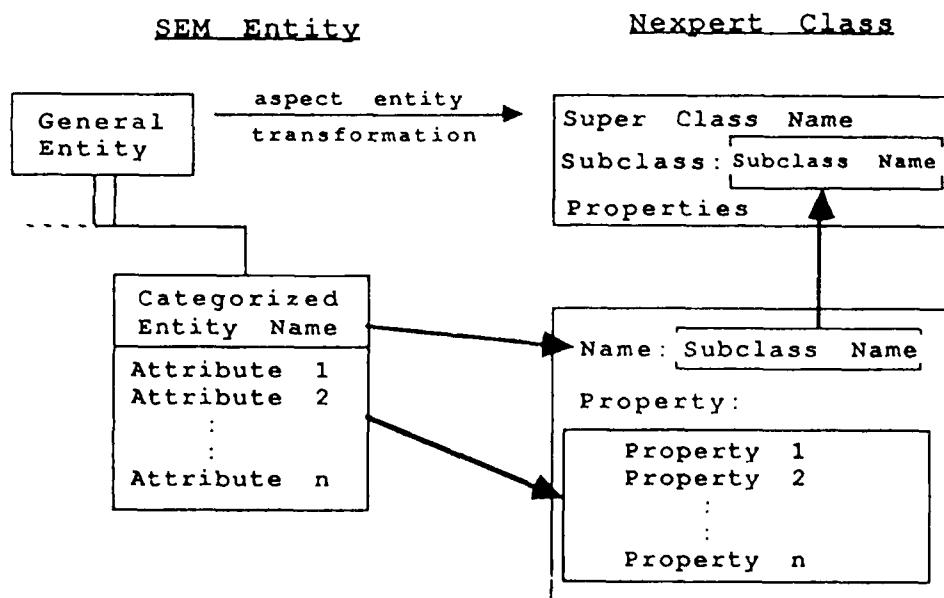
Nexpert's database consists of an object-net (class and object definition) and a rule-net. Entities in the SEM diagram, which is constructed for dynamic structural knowledge, can be directly transformed to Nexpert classes. Inference patterns defined in the SEM diagram are also translated into Nexpert rules. General rules of the transformation (from SEM to Nexpert) are described in the following subsections.

Nexpert object-net design.

An aspect entity in SEM is directly transformed to a Nexpert class as shown in Figure 17a. In this case, the entity name and attributes become A Nexpert class name and properties. However, when an entity is a categorized entity, i.e., a child entity of a specialization, the class name of this entity must be included as a subclass of the generalized class (see Figure 17b). Those classes are easily instantiated through data retrieval from the database



a. An aspect entity.



b. A categorized entity.

Figure 17: Transformation of SEM Entities to Nexpert Classes.

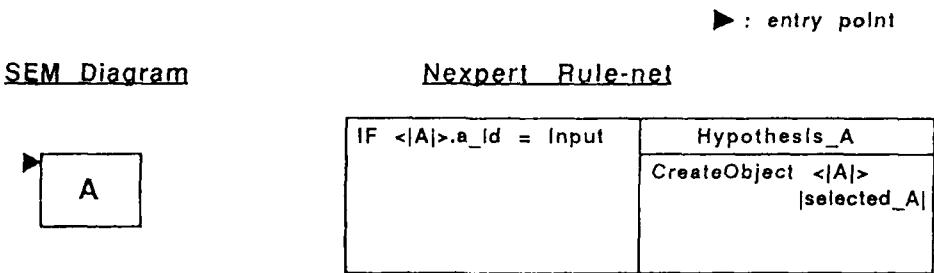


Figure 18: A Single Class Rule-net.

because both Nexpert class definitions and DB schema are coupled through the same SEM diagram.

Nexpert rule-net design.

A rule-net is a realization of inference patterns. The development of rule-net is a complex task and requires a rigorous analysis and design process. Without the help of a structured design method, rule-nets tend to become a mass of 'spaghetti.' Development and maintenance of such a rule-net is very expensive. Although, SEM is not yet a complete structured design method for rule-nets, it provides a semi-structured design methodology to rule-net designers. Transformations of various inference patterns to rule-nets are provided in the following.

Single class: Figure 18 shows the simplest rule-net which involves a single class. In Figure 18, the 'input' is an input datum and the 'a.id' is class A's identifier property. This rule-net instantiates a class-A object whose 'a.id' has the same value as the 'input.' The 'selected_A' is a Nexpert object in which instantiated class A object is stored. Definition of the 'selected_A' may depend on the actual application; however, by default, it can be defined as a subclass of class A.

Subclass: An inference pattern which involves a class and its subclasses can be represented in the same manner as the rule-net of a single class (see Figure 19). In Nexpert, pattern matching in the super class is automatically propagated to its subclasses. However, the

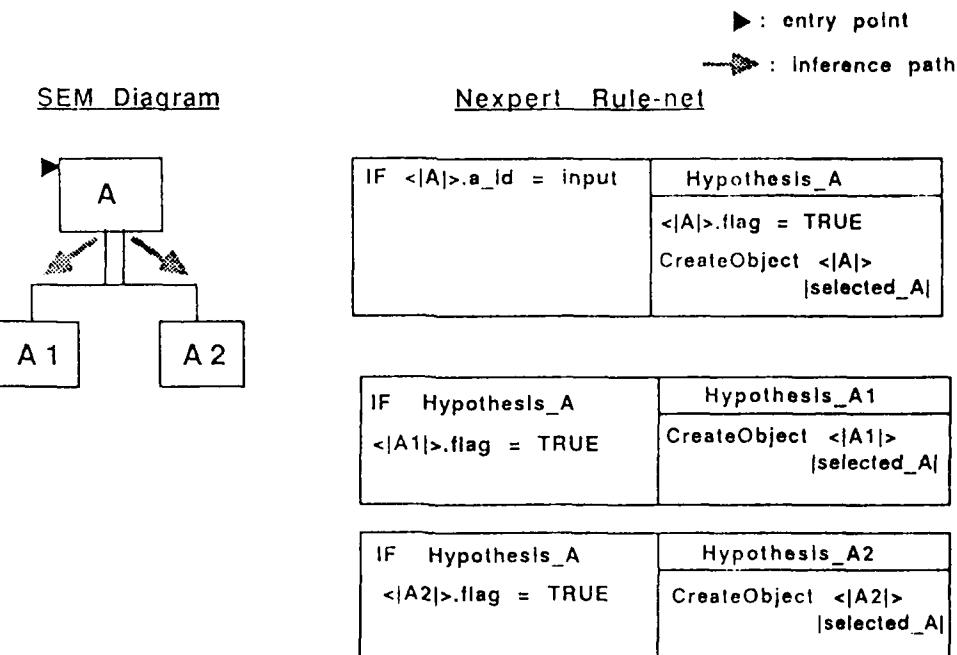


Figure 19: A Subclass Rule-net.

collection of instantiated subclasses must be separately performed as shown in Figure 19. In this figure, the 'flag' is a new boolean property of class A whose initial value must be set to FALSE using meta-slot. The 'selected_A' may contain all properties of class A and its subclasses, although it isn't required to be so, i.e., the selected_A can have only selected properties from the class and subclasses.

One-to-many: The 'input' can enter from the one-side or from the many-side in this inference pattern (see Figure 20). If it enters from the one-side (class A), class B objects are directly instantiated using the 'input' value because class B already contains 'a_id', a foreign key, as its property. If the 'input' enters from the many-side (class B), then the 'a_id' is extracted from the class B object and stored in the 'a_in.' The 'a_in' is a Nexpert object which must be defined when this rule definition is completed. Since 'a_in' is an identifier of class A, a single class A object will be instantiated by the next rule.

Many-to-many I: An inference pattern between two classes which have a many-to-many relationship requires a loop. In Figure 21, class A-B is an intersection class resulting from

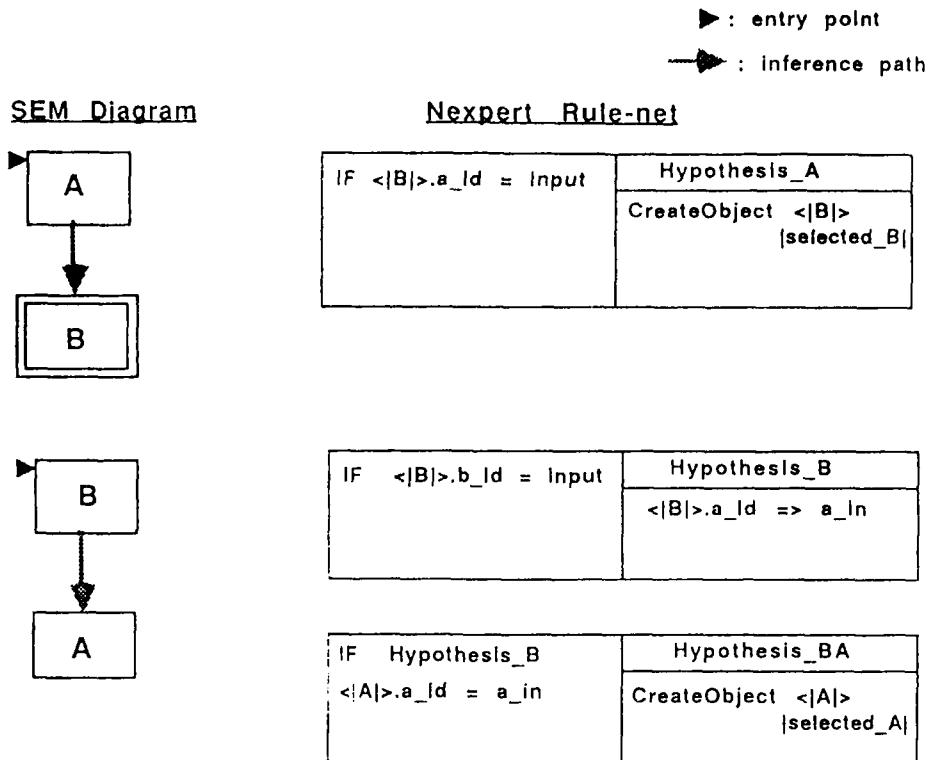


Figure 20: A One-to-many Rule-net.

a many-to-many relationship. The loop starts from the intersection class, A-B, and ends at the goal class, B. To make the loop work, a loop counter is incremented and hypotheses are reset at the loop-end rule which has the Hypothesis_AB. Without this, the loop would be executed once. The loop terminating condition is determined by the value of $\langle |A-B| \rangle.flag$. This condition is checked at the loop-start rule which has the Hypothesis_B. Omission of this process will cause an infinite loop. The class A-B's flag is initialized to FALSE at the class property definition by using a meta-slot.

Many-to-many II: Figure 22 shows two inference patterns in a many-to-many situation. Two separated initial rules and loop-start rules must be prepared. However, the loop-end rule can be used for both inference patterns. In order to use the common loop-end rule, both loop-start rules must have the same hypothesis name (Hypothesis_AB).

Combination I: Figure 23 shows an inference pattern in which the goal class, B, has subclasses, B1 and B2. Pattern matching on subclasses is propagated only if its identifier is used for the pattern matching condition. Thus one extra rule, which has Hypothesis_B, is

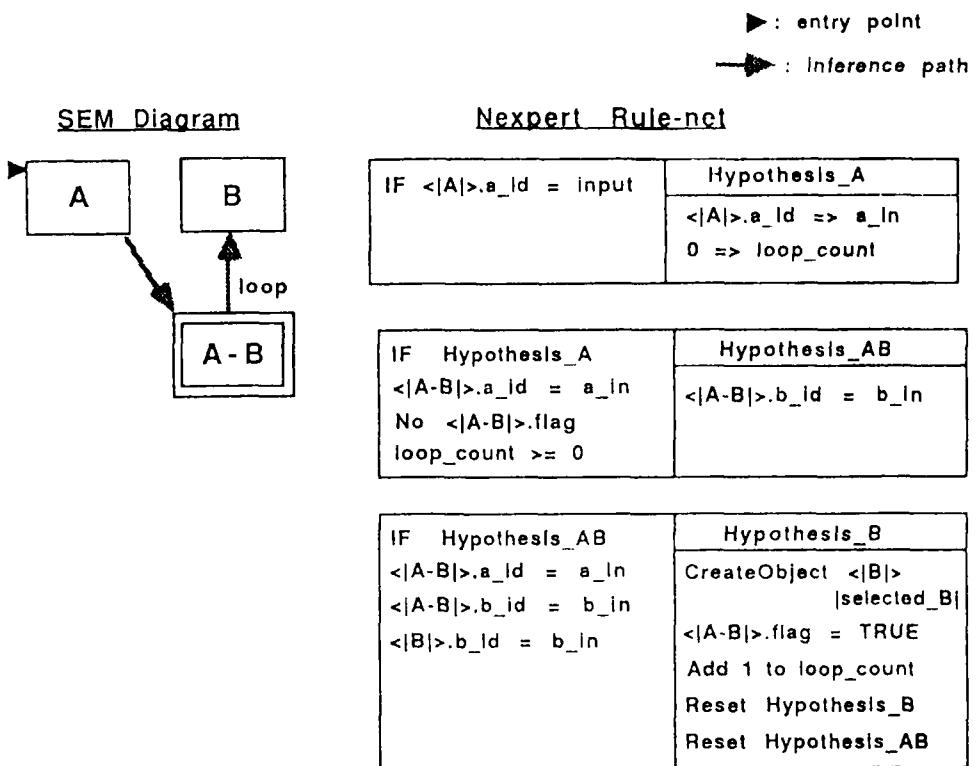


Figure 21: A Many-to-many Rule-net with One Inference Pattern.

used to extract class B's identifier in Figure 23. The instantiation of subclass objects are same as the subclass rule-net.

Combination II: First three rules of the rule-net for the inference pattern depicted in Figure 24 are identical to the aforementioned rule-net for the many-to-many I. However, for this combination rule-net, each subclass object of class B must be collected after the loop is completed.

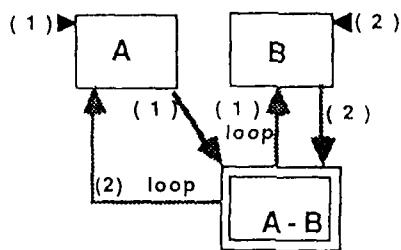
3. Final Comment

After the transformation from inference patterns to rule-net using the SEM methodology, the rule-net has to be tuned. This tuning requires some experience with Nexpert. The use of this methodology does significantly reduce the initial development cost, however, and is also expected to reduce the maintenance cost of the developed KB. Some comparative data of KB development from our experience are summarized in Table 1.

Both examples in this table strongly indicate that we can reduce the size and the development cost of KB by using this methodology. Furthermore, example 1 is extended to

► : entry point
 → : inference path

SEM Diagram



Nexpert Rule-net

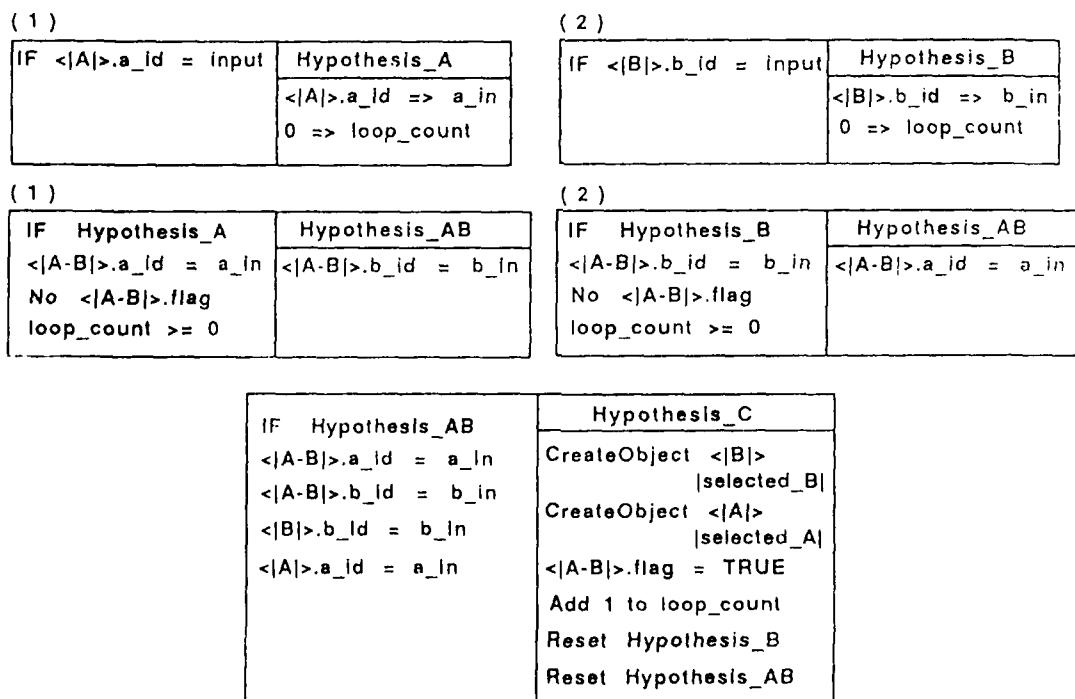
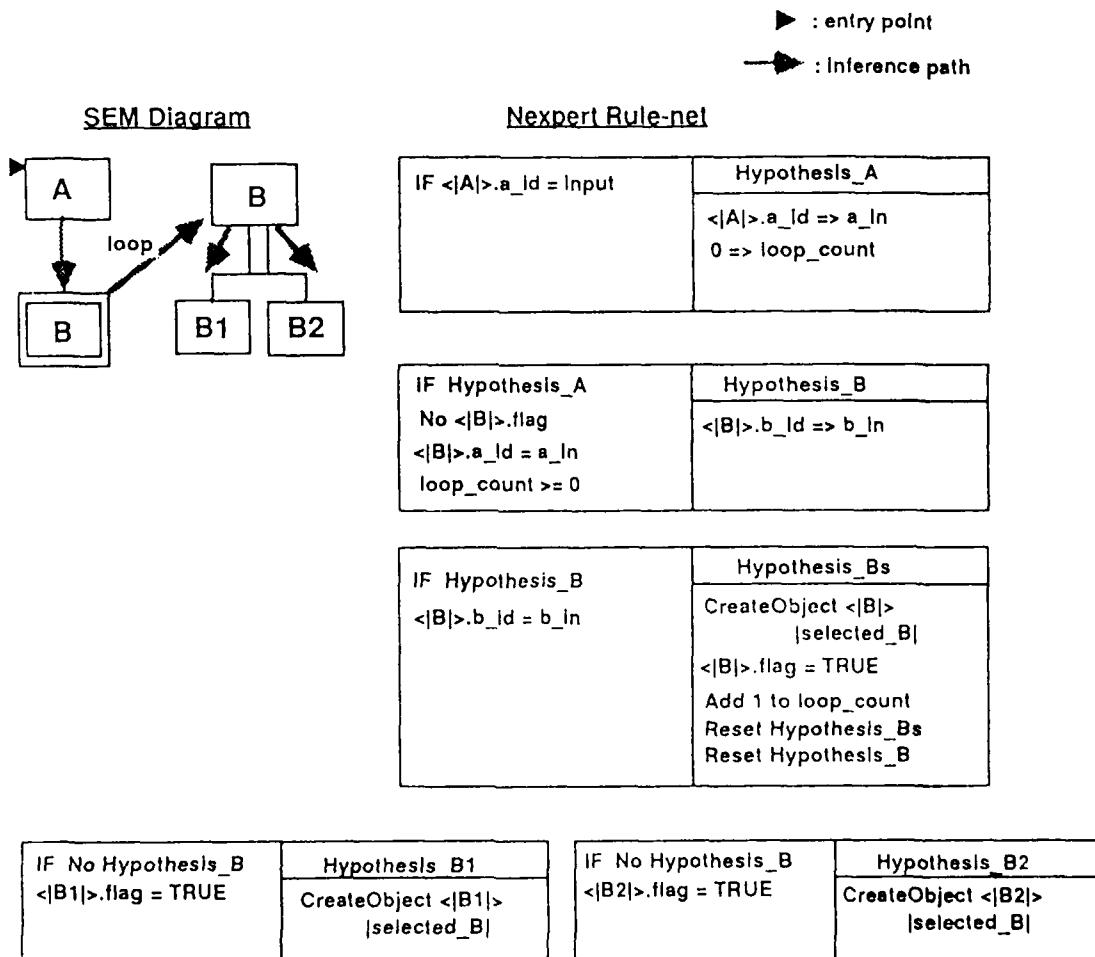


Figure 22: A Many-to-many Rule-net with Two Inference Patterns.



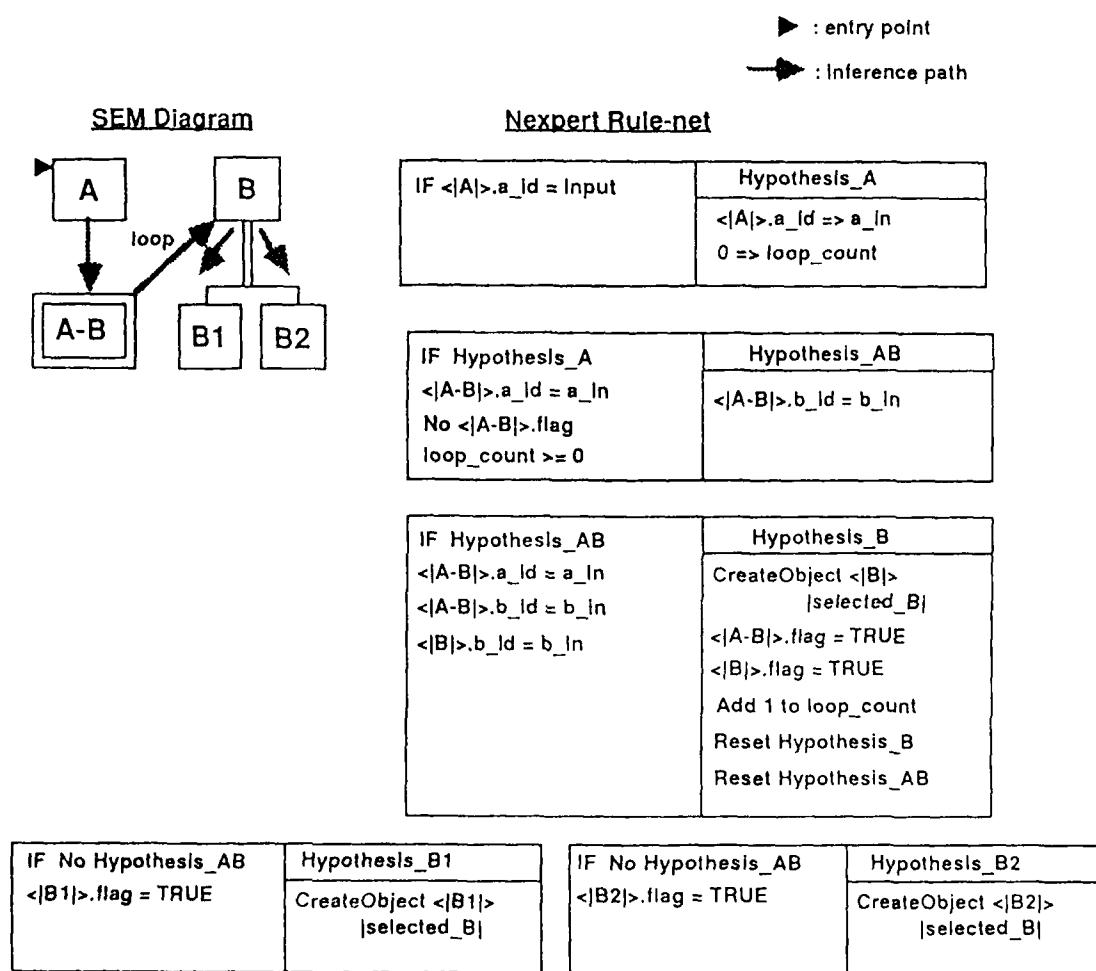


Figure 24: A Combination Rule-net: Many-to-many with Subclass.

Table 4: KB Development with The Methodology vs. without.

Example 1	With methodology	Without methodology
Function	Keyword search (one IP)	Keyword search (one IP)
Rule-net Size	9 rules	14 rules
Development Time	8 hours	20 hours
Example 2	With methodology	Without methodology
Function	Meeting member selection	Meeting member selection
Rule-net Size	12 rules	41 rules
Development Time	5 hours	40 hours

facilitate the aforementioned four inference patterns using this methodology. This modification of example 1 took only 8 hours and 14 rules more. It was estimated to take more than 30 hours and 30 rules if the methodology was not used for this modification.

Development of any realistic and practical knowledge-based system is a non-trivial task. An expert system shell (EES) is supposed to ease this development process. However, typical EES are either too simple to accommodate complex inference patterns or too difficult to use effectively. Nexpert system belongs to the latter type of EES, although it is the most sophisticated EES available. Without the help of the structured design methodology, the use of Nexpert is a major task. However, with the SEM methodology, development and maintenance of a complex knowledgebase can be performed in a cost-effective manner. The development of a distributed DB/KB is also an extremely complex task, and the maintenance of the developed DB/KB will be even more so. Therefore, it is invaluable to complete this methodology before the development of distributed DB/KB takes place in the future IOIS project.

4.4.1 Interface

The interface is being approached from several angles. Menu types and structures are being investigated and compared to more traditional command line approaches. The focus is on text based screens, although graphic-based screens will be used in future prototypes. As much as possible, the interface will appear similar across different software and hardware platforms.

Intelligent interfaces are also being investigated. This has been broadened from simply using a user profile to control access and tailor screens individually, to intelligent interfaces that can provide the most suitable type of help to individual users, improve ease of use, and shorten learning time.

The interface tools provided in the latest version of Nexpert have been improved and are being tested. Nexpert's callable interface allows external C routines to be integrated into an application. Enhanced tools written in C are being developed that will use the callable interface.

4.4.2 Calendar / Scheduler

The Calendar/Scheduler was targeted for implementation because the IOIS design revealed it to be a versatile tool. It can be used in a batch mode or in a stand-alone mode. In the batch mode, it is used by RME to monitor project due dates and by the Session Planner to schedule GDSS sessions. In the stand-alone mode it can be used to schedule meetings, resources or appointments. Although a number of calendar tools are commercially available, none provided the capability to incorporate intelligence or supplied a programming interface that could be integrated with the IOIS. Therefore, the Calendar/Scheduler was designed and developed with the following capabilities:

1. Monitor project activities,
2. Schedule meetings,
3. Specify user profiles (e.g. times convenient for meetings),
4. Schedule resources,
5. Schedule appointments

5 Implementation

The IOIS architecture was implemented in order to demonstrate its feasibility, and to identify issues in porting the architecture to offices with different resources (such as those in a Unix environment, on a network, etc). Only a subset of the office functions have been selected for implementation. These are: 1) AGDSS, 2) ESP, 3) RME, and 4) Calendar/Scheduler.

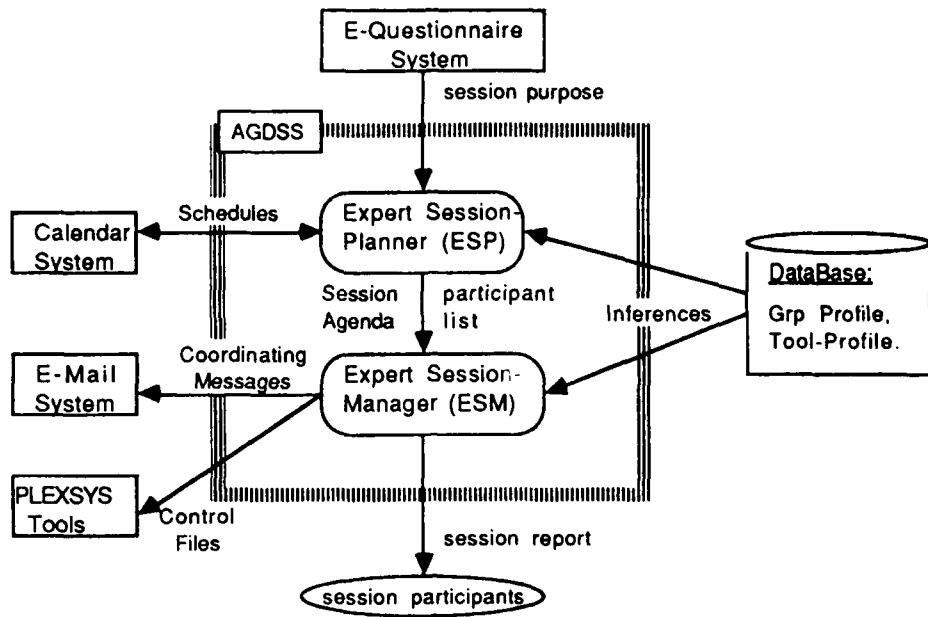


Figure 25: Architecture of AGDSS

5.1 AGDSS

Figure 26 shows the high-level architecture of the prototype system currently being developed at the University of Arizona's Collaborative Management Room. The components shown within the framework are the internal components of the prototype ES, while the external components are either commercial software packages or in-house systems developed for the current suite of GDSS tools. The internal components described in this subsection consist of the tool selection, group selection, and calendar scheduling module.

1. Group Selection Module

The meeting coordinator begins ESP with the group selection module to determine the type of the meeting, the topic of the meeting, and other information. This information is first inferred from the knowledge base which fires appropriate rules and determines more information from the meeting coordinator as it 'back-chains' through the knowledge base. Once it has determined the meeting type and topic, the group selection module instantiates the user profile knowledge from the data base and selects participants who should participate in the meetings based on the facts from their profile. Personnel interests, responsibilities,

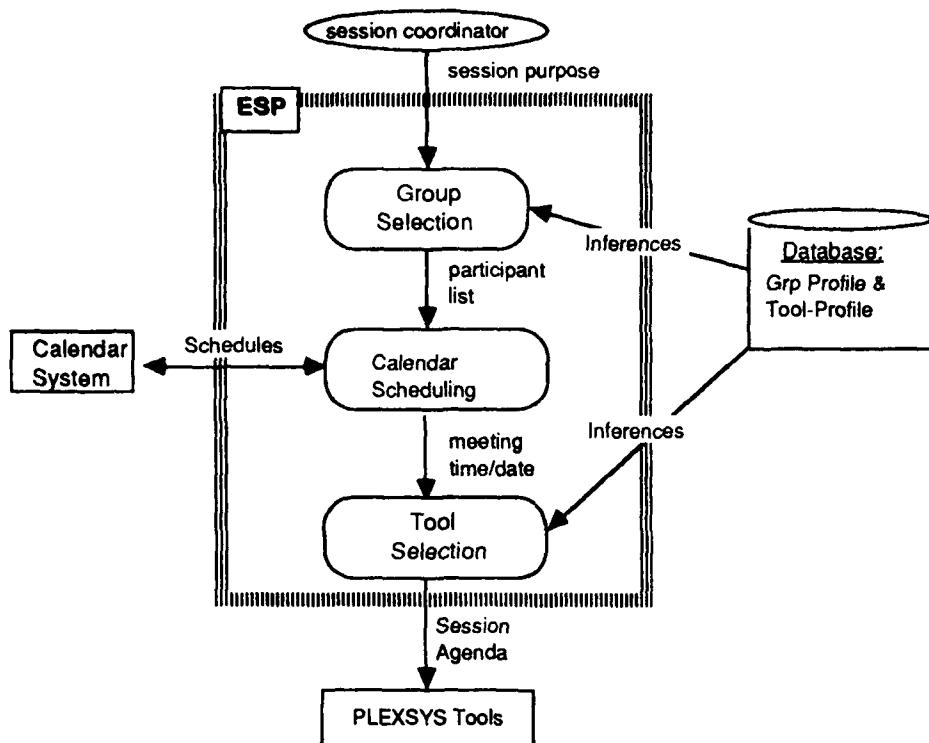


Figure 26: ESP High-level Architecture

and organizational affiliations are all factors determining membership in a potential group meeting.

The meeting topic helps in determining the initial list of participants. For instance, if the topic involves discussion on a project then all members responsible for the project are selected, further the supervisor of the project will be selected so also any member who has listed interest in the area of the project. The meeting type, on the other hand, helps in determining who should be added or deleted from the list. For instance, if it is a preliminary discussion meeting, then the supervisor will be removed, or if the discussion involves financial matters, then the financial officer involved with the project will be added. This process is iterated until the knowledge base is exhausted and a final list is presented to the meeting coordinator. The meeting coordinator can use the list as is or make further changes depending on his preferences. The role of group selection is merely to provide a starting point for the meeting coordinator.

The group determination module is implemented using Nexpert Object, an expert system shell from Neuron Data Inc. The knowledge base consists of 41 rules which select organizational personnel facts from a DBase 3 database, and configure a participant list. This list is then written to an ASCII file for later reference.

2. Calendar Scheduling Module

The calendar scheduling module is strictly used for meeting sessions involving face-to-face interaction. Once the participant list is determined, this module checks the calendar information of each of the participants selected as well as the calendar of the meeting room to arrive at a final time for the session. Currently, the calendar program is simplistic in the sense that it selects the first available time for the participants. Conflict resolution techniques will be included in future implementations of this module.

This module is currently implemented in Turbo Pascal and interacts with a calendar program to access the personal schedules of the participants.

3. Tool Selection Module

Once a group membership list has been constructed and meeting times have been arranged, the meeting coordinator used the tool selection module to determine the GDSS tools required for the session. The coordinator must have already used the group selection module at this point because he is expected to answer questions regarding this group's characteristics. For example, the coordinator must know the extent of the participants' familiarity with the topic as well as the extent of common knowledge among the group members. Additional questions are asked concerning problem characteristics such as whether or not the problem can be segmented and whether or not stakeholder identification is important. Once all of the questions are answered, a list recommended tools is written to a file with their accompanying certainty factors. As with the group membership list, the coordinator at this point is free to modify the recommended list of tools to reflect his own desires.

The module is also implemented in Nexpert Object and has 68 rules. The list of selected tools is written an ASCII file which is referenced later by the group facilitator.

5.2 ESP Implementation Status

ESP is implemented on an AT&T 6386 WGS workstation. The prototype has met with some initial success in knowledge base validation. The knowledge for the tool selection module was acquired through extensive interviews with local expert facilitators. The knowledge has initially been represented in the form of IF-THEN rules with the possibility of representation in the form of frames in future versions. Since the selection of tools is primarily a design problem, a forward-chaining inferencing strategy is used. Further, 20 user-profiles are stored in the database, implemented in (Dbase III+) and loaded at run time by the prototype system.

The prototype system is implemented using Nexpert Object in the DOS environment. This environment satisfies many of the implementation criteria by providing graphic interfaces, portability among DOS, Vax, and Unix products; excellent database and knowledge base support, and external program calls.

ESP has been pilot-tested with case studies and field experiments. Initial test results have been positive as the tool has proved to accurately match the expert's prescriptions for group and tool selections. However, users have expressed some discomfort using the tool selection module due to the high number of questions asked prior to a recommendation: currently 14. Additional work is being conducted to add additional rules while structuring the line of questioning, thus reducing the total time to complete the questionnaire.

Further enhancements, including intelligent support for the facilitation stage (Expert Session Manager) which uses the output from ESP to automate the facilitation process, are being designed.

In summary, Huber [1] notes that the success of a particular EMS depends on the range of tasks supported and its frequency of use (both of which are mutually dependent). Through the added functionality and flexibility of a chauffeured, asynchronous, distributed GDSS, ESP provides a sufficient number of tools and adequate support to achieve the critical threshold necessary for a high frequency of use.

Huber also states that effective facilitation depends upon:

1. Technical Competence:

The facilitators and group participants are assumed to already be familiar with each of the GDSS tools in the agenda. ESP provides effective support by choosing the most appropriate

tool for a given task and group.

2. Knowledge of the Planning Process:

The prototype satisfies this requirement through accumulated knowledge of the requirements, purposes, and goals of the group to be facilitated.

3. Group Facilitation Skills:

ESP does not address this issue, but the Expert Session Manager currently under development will send out reminders and notices while performing many of the duties of a human facilitator.

The type of organization, its methods and goals, and its purposes for the session are all important for the facilitator (human or automated-chauffeur) to know. Knowledge about the initiator and members participating is also important. The prototype system meets all of these requirements.

5.3 Implementation of RME

The Resource Management Expert (RME) provides assistance to AIRMICS personnel in all activities involving resources: funds/expenses, personnel, equipment, and schedules. The process that AIRMICS goes through to plan, execute and evaluate research in the areas of interest consists of selecting the projects to be funded, collecting the resource information, allocating resources, and administering contracts. The implementation of RME consists of five components: 1) a keyword help facility for assisting in the DTIC search, 2) a forms oriented facility for completing the BAA forms required for preparing the acquisition package, 3) A personnel allocation facility, 4) a database query/update component for viewing/modifying the status of resources, and 5) a calendar/scheduler facility for monitoring schedules and due dates.

Four components of RME have been implemented: keyword help, retrieval of project histories, personnel allocation, acquisition package preparation, and resource status/update. In addition, the Calendar/Scheduler, an office tool accessed by RME and other IOIS components, has been implemented.

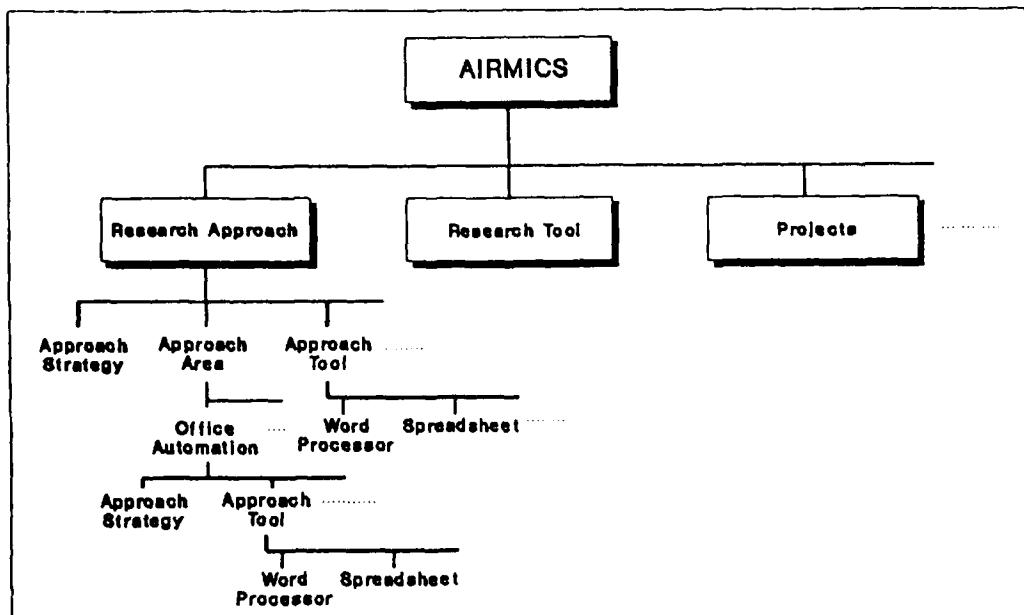


Figure 27: The Keyword Structure

5.3.1 Keyword Help

After a project has been approved for funding, the project manager or other person responsible for preparing an acquisition package accesses EIS (outside of the AGDSS setting) to retrieve information on related projects that are in progress or have been completed in the past. Two sources of information are available: the DTIC database, and the local AIRMICS database containing information on past AIRMICS projects. (Presently, only the DTIC database is computerized.) A keyword help facility provides information on related words that could be used for database queries. Keywords can be browsed alphabetically or by relationships between words. A complex tree structure allowing multiple parents and children is used to map the relationships between keywords; part of this structure is reproduced in *Figure 27*.

Keywords can be browsed from the top level of the tree downward or from any known point within the tree with the starting keyword entered at a prompt. Keywords at each level are displayed in pop-up menus similar to other menus in the interface. The tree can be searched upward or downward by selecting any of the keywords listed in the menu. After finding the related keywords, the system can automatically generate a query to the DTIC database for information, or the user can choose to manually create or modify the query. Project information is returned in

the same format as used in other modules. AIRMICS queries will be similar when an electronic database is implemented.

5.3.2 Retrieve Project History

5.3.3 Retrieve Project History

This module is designed to allow users to write queries to the DTIC and local AIRMICS database. It can be used by itself or in combination with Keyword Help. If a user chooses to write a query, a simple on line help facility will provide the needed syntax. If the user is not sure what keywords should be used in a query, Keyword Help will be accessed first. Keywords can be browsed through and chosen while in Keyword Help. The user can then view the words that were chosen in Keyword Help while in the Retrieve Project History module. Words chosen in Keyword Help can be selectively removed or added to the list in Retrieve Project History before sending the query. If this option is taken, the system will also take care of formatting the query.

The results of a query will be first presented as a list of projects meeting the query conditions. A user will be able to scroll through the projects and call up more detailed information about a specific project by highlighting it and then selecting it (pressing the enter or return key). This is similar to what is often referred to as hypertext in current literature.

Potential users of Retrieve Project History include managers who need more information about past and current projects before making comments in an AGDSS session and officers evaluating potential new projects.

5.3.4 Personnel Allocation

Once a project is selected as a funding candidate, resources are allocated to the project. The allocation component draws the attributes of the project and attributes of the resource from the database similar to the design used in the Information Center Expert (ICE) developed at the University of Arizona's MIS Department [7]. RME invokes the knowledge base and begins reasoning on the project and resource attributes and arrives at a match factor (a number between 0 and 1) which is a degree of fit between the project and resource. For each project, the resources and their matching factors are listed in separate windows (see *Figure 28*). A retrospective explanation facility provides an explanation on how the matching factor was derived. The final

PROJECTS AFMIS VLDB IOIS ISDN SARA	PERSON Hengst Aiken Evans Peters Everett	ASSIGN EXPLAIN
---	--	---------------------------------

ASSIGNMENT AFMIS VLDB IOIS Hengst ISDN	
--	--

Please enter <CR> to assign person to project, <ESC> to quit

Figure 28: Resource Allocation

allocation is done manually by either the division chief or the project officer at AIRMICS. *Figure 29* illustrates some of the rules of personnel allocation.

5.3.5 Prepare the Acquisition Package

A forms facility for the BAA process was implemented using "Vermont Views." Currently two forms have been implemented: the project evaluation and the Purchase Requisition and Commitment (PR & C). Features of the implementation include virtual paging to display forms of any length, ability to link the form with a database to store and retrieve form fields, ability to link multiple related forms together so that they are invoked one after another, context sensitive help, menu choices for filling certain fields (e.g for division, the choices are ACTID, ADMIN, CNSD, and CISD), and automatic field filling.

5.3.6 Resource Status/Update

The Resource Status/Update component is a forms-database oriented facility for obtaining the status of resources and updating new resources. It has been implemented according to the principles of form management described by Tsichritzis [6]. A number of different templates

<u>Condition</u>	<u>Action</u>
If Yes find_person's_edu_level and is proj.edu_req "MS" and is person.edu_level "MS"	mf_edu_level_found Do 1 mf1
If Yes find_person's_edu_level and is proj.edu_req "PHD" and is person.edu_level "MS"	mf_edu_level_found Do 0.8 mf1
Where: find_person's_edu_level	is a hypothesis set to true in a previous part of the inference to trigger firing of educational level matching rules.
proj, person	are objects containing the properties of projects and personnel, one at a time
mf1	the match factor for matching educational level of person with educational level required for projects.
mf_edu_level_found	is a hypothesis set to true when mf1 is found.

Figure 29: Personnel Allocation Rules

Project Name	Approved?	Awarded?	Time	Cost	Source of Funds

Table 5: RME Input Form

allow the user to view and update the resource information. Whenever a user belonging to a particular project fills an equipment request form or a travel form, the expense is automatically recorded against the project.

RME will also be used to gather and consolidate project information during meetings. During various meeting stages, RME input forms would be accessed and updated by participants. General information would include project names, times, funds needed, and sources of funds. A spreadsheet format is used with data appearing in highlighted cells. The column headings are illustrated in *Table 5* and the rows represent separate projects. A function key brings up a second form with more detailed information on the project currently highlighted. Detailed information about the project would include the name of the project manager, review schedules, project descriptions, etc.

The first form will also display information about current projects that have been entered

in prior meetings. If the projects are either approved or finalized, an appropriate box will be checked. Checked projects cannot be modified during a group meeting. RME can be accessed independently from the meeting tools to make the modifications.

5.4 Calendar/Scheduler

The IOIS architecture provides support to different levels of workers through a combination of intelligent systems as well as with conventional tools. One of these support tools, the Calendar/Scheduler, is a versatile tool that can be used in a batch mode or in a stand-alone mode. In the batch mode, it is used by RME to monitor project due dates and by the Session Planner to schedule GDSS sessions. In the stand-alone mode, it is used to schedule meetings, resources, or appointments. Although a number of calendar tools are commercially available, none allows incorporating intelligence or supplying a programming interface that could be integrated with the IOIS. Therefore, the Calendar/Scheduler was designed and implemented with the following capabilities:

1. The ability to monitor project activities,
2. The ability to schedule meetings,
3. The ability to specify user profiles (e.g. times convenient for meetings),
4. The ability to schedule resources,
5. The ability to schedule appointments

The prototype Calendar/Scheduler was implemented in Turbo Pascal 4.0 and can be called from Nexpert.

6 An AIRMICS Scenario Illustrating the Use of IOIS

The tasks to be supported with IOIS can divided into two general categories: 1) Planning for Projects, and 2) Initiating and Monitoring projects. The technologies to be used for these tasks can be subdivided into three general categories: 1) AGDSS (Asynchronous Group Decision

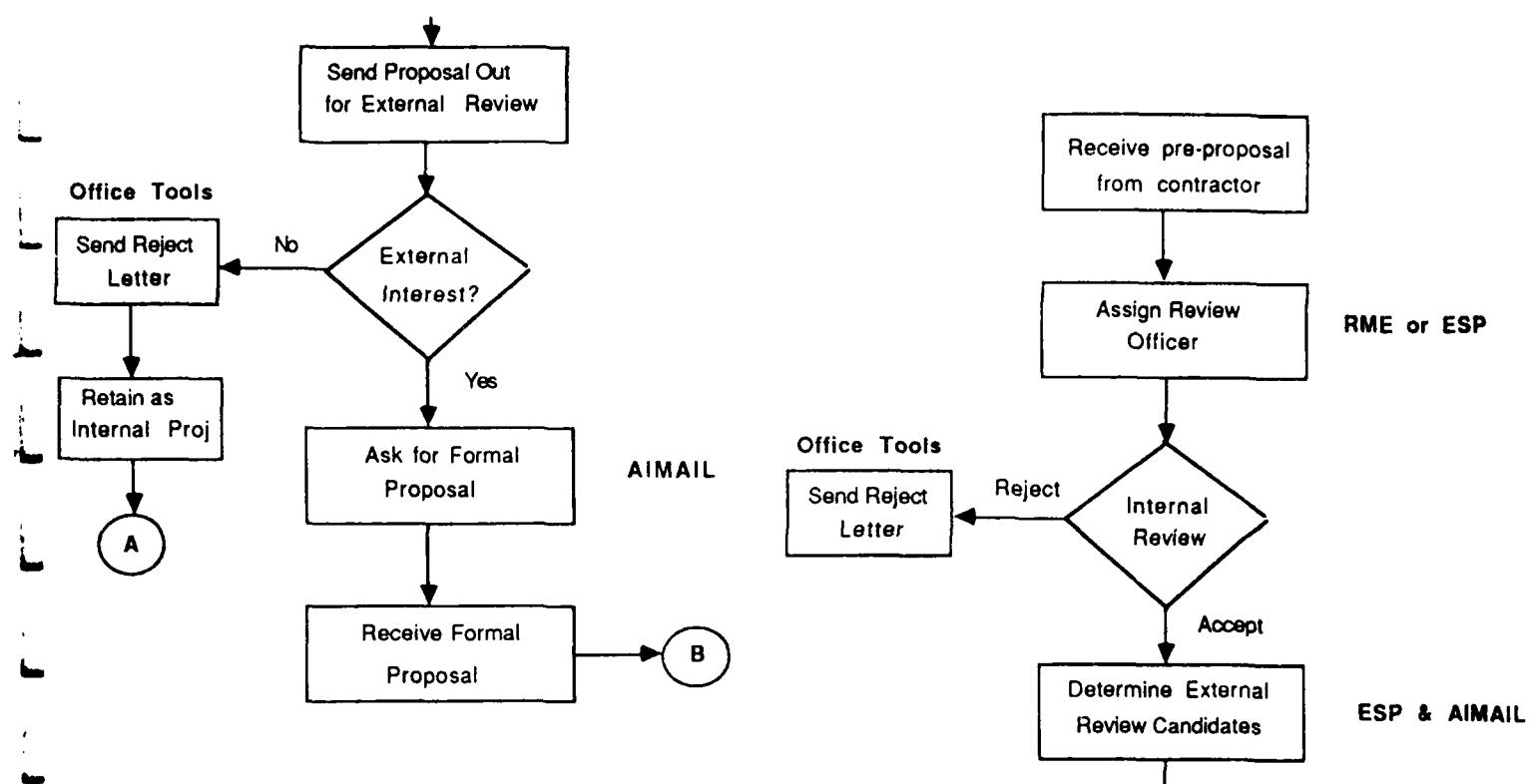


Figure 30: An AIRMICS Scenario Using IOIS

Support System) for generic group problem investigation and solution, 2) RME (Resource Management Expert) for office resource management, and 3) Various commercial software tools which support simple tasks. The manner in which the tools support the RDTE process is discussed in this section. There is some planning involved in deciding on the areas of interest, but this will not be discussed.

The process starts with receiving a pre-proposal from the contractor (*Please refer to figure 30*). A review officer is assigned to the project proposed by the contractor. At this stage either the personnel allocation model in RME or ESP could be used to make the assignment. RME makes use of the expertise requirements of the project, the availability and expertise of the researchers to recommend a suitable project officer.

The proposal is reviewed internally by the director and division chiefs to determine whether or not it is consistent with AIRMICS RDTE plan. The proposal is then either accepted or rejected. If it is rejected, a letter is sent to the contractor proposing the project. The IOIS architecture provides access to Office Tools and a word processor is used to prepare and send the rejection letter.

If the proposal is accepted by AIRMICS, it is subjected to more evaluations. But at this stage,

AIRMICS also attempts to solicit external sponsorship for the project. Suitable candidates for the project are selected using ESP. Criteria such as the interests and current projects in progress are used in this decision. The proposal is then sent out for the review. AIMAIL could conceivably be used for both identifying the candidates and mailing the proposal. But at present this has not been adapted for this purpose.

If there is no external interest for the project, then a rejection letter is sent again to the contractor, but since the project was already found to be consistent with AIRMICS' objectives, it is retained as an internal project. A rejection letter is sent to the contractor. As in the internal decision process, Office Tools are used for sending the reject letter. If there was external interest for the project, the contractor is asked to submit a formal proposal, including budgets, deliverables etc. Again, AIMAIL could be used for communicating this information.

When a formal proposal is received from the contractor (*Please refer to figure 31*), it is subjected to another internal review and another external review. Again, the outcome of these decisions may be to reject the project (a rejection letter is sent out with the help of Office Tools), or to accept it. At this stage, resource requirements of the project are assessed with RME. It makes use of the knowledge base to estimate costs for the project. The knowledge base contains heuristics from previous projects, such as the time it takes to develop communications software or the time it takes to analyze a potential system. A final high level meeting takes place involving the director of AIRMICS and others from ISC and ISEC. The Expert Session Planner (ESP) enables the project officer to plan the meeting in terms of selecting the group participants (based on organizational knowledge), selecting the GDSS tools (based on tool knowledge), and determining the time of the meeting (based on participant's calendar). After determining the meeting participants with the help of ESP, mail messages announcing the meeting are sent automatically with AIMAIL. The Voting tools of PLEXSYS are used in the APRC meeting to vote on the ultimate fate of the project. If approved, the project's status is updated in RME, to trigger other activity relating to monitoring the project.

Once a project is finally approved, the project officer prepares an acquisition package to acquire the services of the contractor. Two tasks are supported in preparing the acquisition package: searches of databases for historical project information and filling the forms associated with the acquisition package. The COR accesses RME (outside of the AGDSS setting) to retrieve

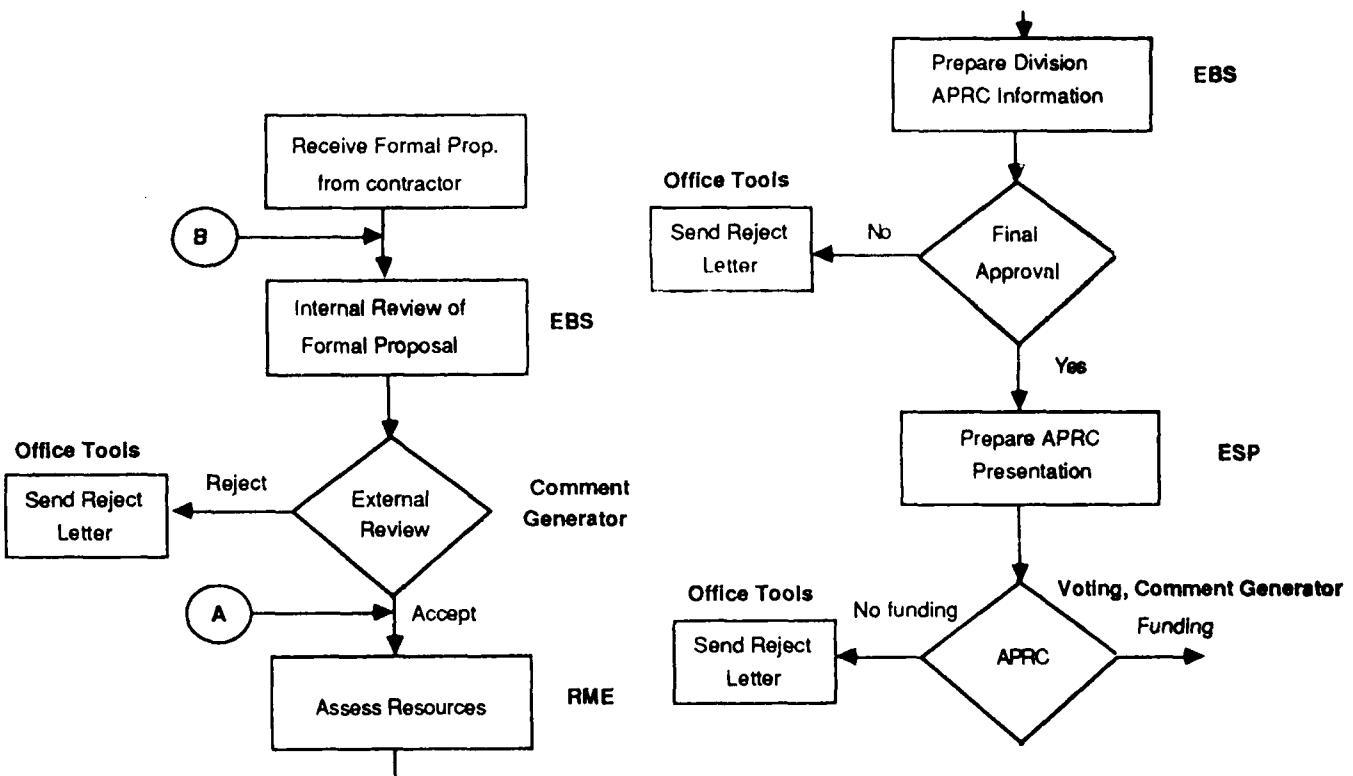


Figure 31: AIRMICS Scenario Contd..

information on related projects that are in progress or have been completed in the past. Two sources of information are available. One is the DTIC database. The other is the local AIRMICS database containing information on past AIRMICS projects. In either case, a keyword help facility will provide information on related words that could be used for database queries. After finding the related keywords, the system can automatically query the local AIRMICS database for information, or the user can choose to manually create the query. DTIC queries are similar. Support for filling the BAA forms online is provided. Some of the fields in the forms are automatically filled from information contained in the database (such as the division, addresses etc.). After the package is prepared, it is forwarded to the contracting officer at Ft. McPherson. The contracting officer conducts a financial analysis of the contract and either approves it or suggests changes. If there are any changes, they are made by the Contractor and co-ordinated by the COR. The project is then formally awarded.

After the acquisition package is finalized and processed, personnel and equipment are formally allocated to projects (not shown). Personnel allocation is used for assigning CORs and researchers to projects. It invokes a knowledge base to determine the degree of fit between the project and the personnel.

Once a project is under way, it is monitored primarily by the COR and his division chief. The COR may travel to the site of the contractor for which it may be necessary to complete some paperwork. The purpose of the COR's visit is to assess the progress of the project and to recommend or approve any changes if necessary. RME provides support for reminding the COR of the project review schedule and for completing the paperwork. After the COR completes the review he makes use of the office tools to complete his trip report. Also, when a project is in progress, and especially for internal projects, the COR may occasionally check the resource status to see if their usage is in line.

When the project terminates, a final IPR is conducted. The director, the division chief, the COR, and any interested party within AIRMICS participate in the review. The meeting participants, the tools required, and the meeting times are all planned with ESP. AIMAIL (AGDSS) is used to distribute the results of teh research. After the final IPR, the COR updates the DTIC database and performs some additional internal procedures to terminate the contract.

This section illustrated one possible scenario involving the use of IOIS tools. The architecture of IOIS permits it to be used for a number of purposes. It is also general enough to be used for other types of offices.

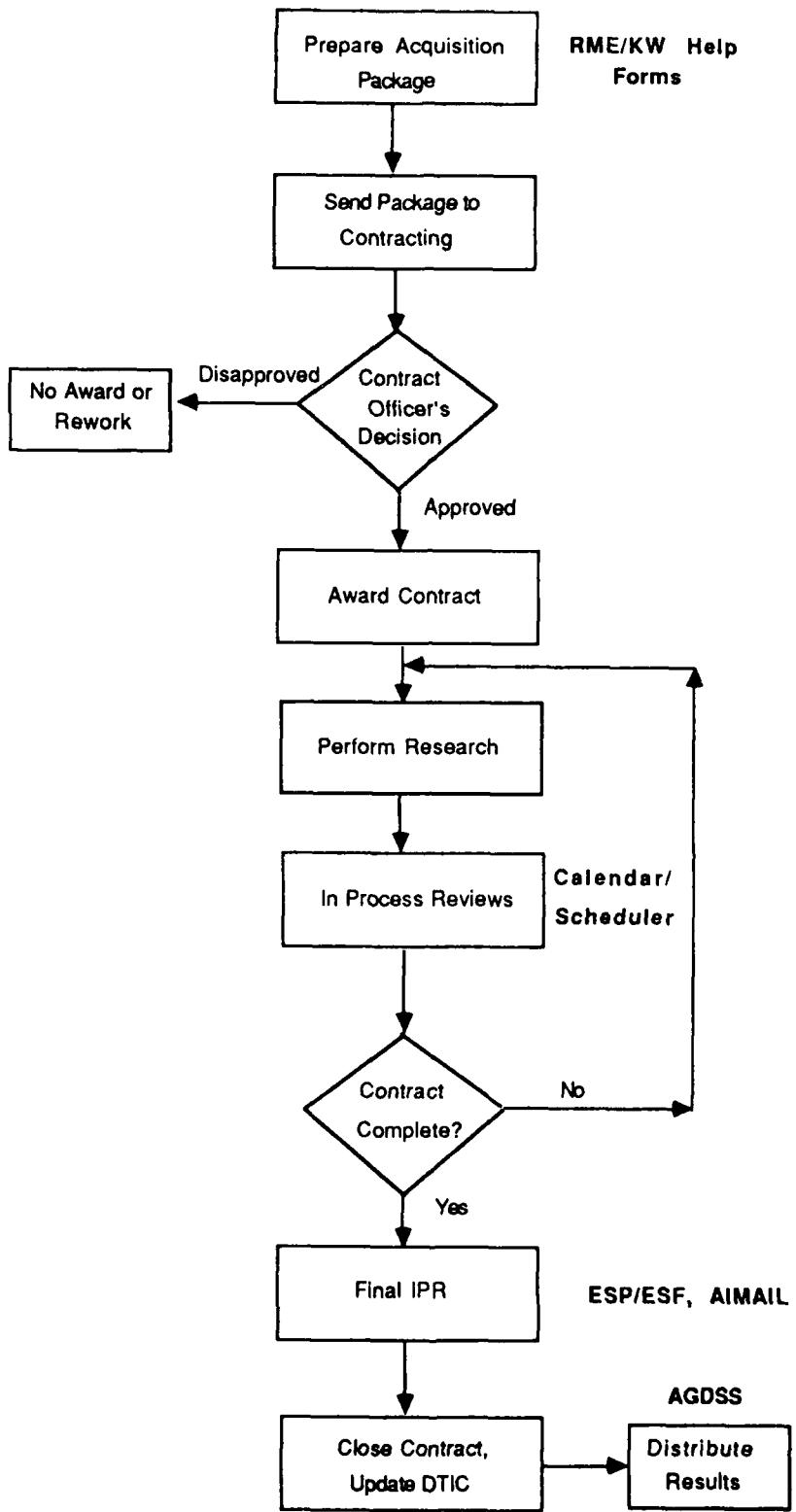


Figure 32: AIRMICS Scenario Contd..

References

- [1] Huber, George P., "Issues in the Design of Group Decision Support Systems," *MIS Quarterly*, September 1984, pp. 195-204.
- [2] Liu Sheng, O., Motiwalla, M., Nunamaker, J., & Vogel, D., "A Framework to Support Managerial Activities Using Office Information Systems," *To appear in Journal of Management Information Systems*.
- [3] Nunamaker, Jay F., Amaravadi, Chandra S., Motiwalla, Luval F., "OSA: An Office Systems Architecture for Supporting Managerial Activities", *U.S. Army Information Systems Engineering Command's Technology Strategies Conference*, February 1988.
- [4] Software Engineering Research Center, "An Information Systems Plan for the Army Institute for Research in Management Information, Communications and Computer Science," *Internal Report* Georgia Institute of Technology, Atlanta, Georgia, June 1988.
- [5] Software Engineering Research Center, "Information Systems Analysis for the Army Institute for Research in Management Information, Communications and Computer Science," *Internal Report* Georgia Institute of Technology, Atlanta, Georgia, June 1988.
- [6] Tsichritzis D., "Form Management," *Communications of ACM*, 25(7), July 1982, PP 453-478.
- [7] Vinze, A.S., Heltne, M.M., Chen, M., Nunamaker, J.F., & Konsynski, B. R., "A Knowledge-Based Approach for Resource Management," *8th Annual ICIS Conference*, Pittsburgh 1987.